nited States
epartment of
griculture

Economic
Research
Service

National
Economics
Division

August 1983

The Automated Weather/Yield System

Michael D. Weiss

THE AUTOMATED WEA Economics
Division, Economi e,
Washington, D.C.

This document des the
Automated Weather ecasting
system maintained of the
system's applicat ed.

Keywords: Agricultural climatology, climate, crop-yield forecasting, computer systems, information systems, Speakeasy, weather.

* *
* This paper was prepared for limited distribution to the research *
* community outside the U.S. Department of Agriculture. *
* *

ACKNOWLEDGMENTS

The author wishes to thank Gwen Powell and Dorothy Armentrout for typing the manuscript and providing other valuable assistance in the preparation of this report.

CONTENTS

FIGURES

Page

Introduction

The winter of 1976-77 brought with it not only extreme cold, but also the threat of drought in significant portions of U.S. crop-producing regions. This situation led to a heightened awareness of the importance of weather for U.S. agriculture. With this backdrop, the author was charged with the task of developing improved operational forecasting models reflecting quantitatively the impact on crop yields of "weather-up-to-the present" at successive stages of the crop year. The models were intended to assist commodity specialists in their economic analyses and to serve as one element of a comprehensive, computerized, cross-commodity forecasting system.

It was apparent from the outset of this research that, because of the large amount of weather data needed for the development of the models, the computer would play a major role in storing and manipulating data. What was not then apparent but became so as the work progressed was that the computerized weather/yield system which was to evolve from this particular project would be sufficiently versatile and general to enable it to serve as a broadly applicable resource not only within the Department of Agriculture (USDA), but in the fields of agricultural and agricultural-economic research generally. Accordingly, a key aim of this work is to acquaint other researchers with the design, operation, and potential of this system, the Automated Weather/Yield System, now in operation in USDA.

Chapter 1 serves as a general introduction to the System. It should provide the reader with a "feel" for what the System is and what it can do. Chapter 2 provides a detailed technical analysis of the System. It is designed to serve a threefold purpose: first, to provide insight into the design of the System at a level of technical detail sufficient to allow researchers at other locations

to install similar systems, if they desire; second, to provide points of guidance to the System operator; and third, in conjunction with the Appendix, to provide a formal record of the Speakeasy software underlying the System.

CHAPTER 1: AN OVERVIEW

The Automated Weather/Yield System is a computerized weather information system with capabilities for econometric forecasting of crop yields. It contains large data banks of monthly weather data and annual crop data and has the ability to extract, combine, and manipulate these data in a highly flexible manner. While the System is basically designed around monthly weather data, it has the ability to monitor weekly temperature and precipitation during the crop-growing season. This ability is particularly useful for crop-yield forecasting.

It is useful to conceive of the System as consisting of four main parts: (1) the weather data bank, (2) the economic data bank, (3) the data-manipulation subsystem, and (4) the automated forecasting subsystem. This chapter will discuss each of these components in turn.

The Weather Data Bank

Considerations in the Choice of Data

Several considerations entered into the early decision as to what type of weather data to employ in the System. Of paramount importance was the fact that the data would have to be transformed into "weather variables," based on both historical and current weather, for use in multivariate regression equations designed to forecast annual crop yields. Because of the need to preserve degrees of freedom, only a few weather variables could be employed in each equation. Thus, weather data representing a large crop-producing region and covering the many months of the growing season would have to be telescoped into a few aggregate figures. This suggested the choice of data which was already spatially and temporally aggregated.

Another consideration was cost. The expense involved in acquiring and storing the historical records of weekly or more-frequent data would have been prohibitive.

A final consideration was the desire to obtain a collection of historical data which was complete, reasonably "clean," and logically arranged.

NOAA Climatic-Division Data

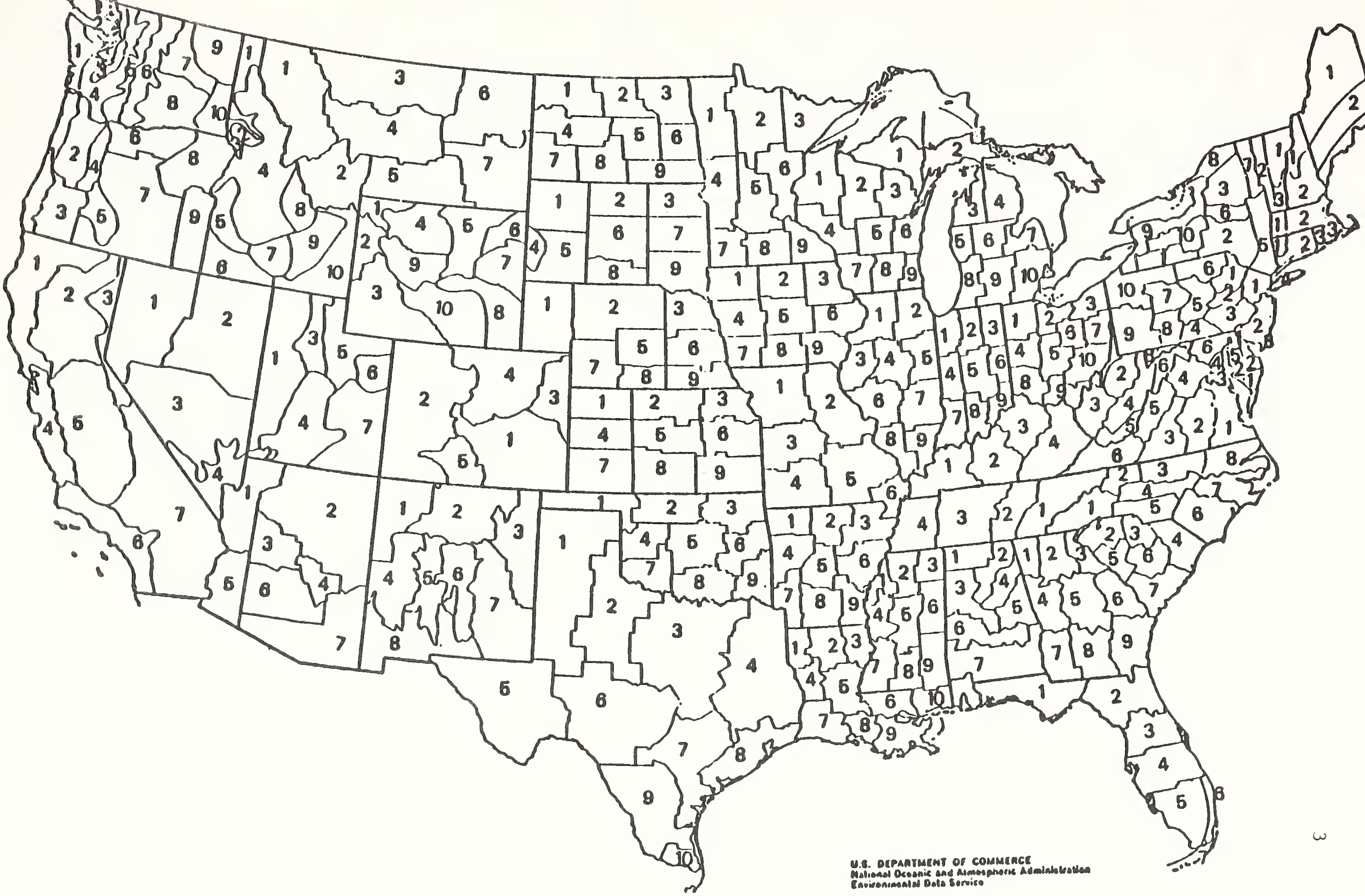
A collection of weather records fulfilling these requirements was found to be available from the National Oceanic and Atmospheric Administration (NOAA) (cf. [1]) and was chosen as the basis of the weather data bank.

The NOAA data is based on a partitioning of each state (and certain other territories) into "climatic divisions"--geographic regions within which the climate is relatively homogeneous. For the 48 contiguous states, there are 344 climatic divisions (Figure 1); many of these coincide with Crop Reporting Districts used by USDA's Statistical Reporting Service (SRS).

The weather data bank consists entirely of NOAA climatic-division data and other records derived therefrom. It currently contains monthly records of (average) temperature and (total) precipitation for each climatic division and state. The climatic-division data extends from 1931 to the present, while the state data spans the period 1931-80 (see [11]). Also included in the bank are a (mostly complete) collection of Palmer Drought Indices for climatic divisions (1931-75) and a complete set of monthly temperature and precipitation records for each of the ten farm production regions (1931-80) (see [11]). In addition, the bank contains 30-year monthly normals, as well as some weekly records (stored temporarily for use in the weekly updating process, as explained below).

The data are stored on disk in an IBM computer operated by USDA's Washington Computer Center. The records for climatic divisions reside in the Economic Research Service (ERS) Time-Series Data Management System TDAM (see [8]),

Figure 1--Climatic divisions of the United States



where they are formatted as monthly or weekly time-series, each series containing the temperature, precipitation, or Palmer Index records for a single climatic division.

The Updating Process

The temperature and precipitation climatic-division records are kept current by a systematic updating process. At regular intervals, NOAA's National Climatic Center in Asheville, North Carolina furnishes ERS with computer card decks containing recent temperature and precipitation records for all climatic divisions. Monthly records are furnished each month throughout the year and cover the month ending six weeks prior. Weekly records are furnished each week during the growing season (approximately March to November) and cover the previous week. Within hours after the receipt of a card deck, its data have been read into the data bank.

The main function of the weekly data is to permit the timely updating of the current month's records. This updating process operates as follows: Within the data bank, each climatic division is assigned a temperature and precipitation "record" for each future month. The assigned "record" is merely the 30-year normal temperature or precipitation for that month in that climatic division. When, ultimately, data are received covering the first week which intersects a given month, those weekly data are used to revise the monthly record, in effect replacing between-one-and-seven days of normal weather with actual weather. When the next week's data are received, seven more days of normal weather in the monthly record are replaced by actual weather. This process of weekly replacement continues until the monthly record consists entirely of an aggregation of actual (weekly) data. NOAA's weekly weather data, however, are intended only to be provisional and are subject to revision.

Thus, six weeks later, when NOAA's "official" record for the given month is received, that record permanently replaces the "aggregated" record just described (Figure 2).

The logical requirements of an automated, weekly, recursive replacement process such as the one just described are more complex than at first might be assumed. On the one hand, one desires that the controlling software be compact, logically simple, and unrequiring of internal change once installed. On the other hand, the replacement process must take into account the fact that different months have different lengths, and leap years affect the length of February; that there is no prior certainty as to in which week in a crop year NOAA will begin publication of its weekly data; that some weeks intersect (and thus must trigger revision of) two months; importantly, that the calendar, and thus the entire interweaving of weeks and months, changes yearly; and, finally, that weekly temperature and precipitation figures must be aggregated into monthly figures in different manners, since temperature is in general expressed as an average, precipitation as a sum.

Despite this labyrinth of detail, essentially a single, simple equation was developed which, applied recursively to each climatic division, accomplishes the weekly replacement process (Figure 3). In this approach, the underlying algorithm--essentially an abstract "model" of the replacement process--remains unchanged through different weeks, months, and years. The only notable requirement placed on the user is that, at the beginning of each crop year, he submit the simple "intersection matrix" INTMAT whose k, j element is the number of days common to month k and NOAA week j. (Actually, INTMAT can be generated automatically, once the user has designated which week is NOAA's "Week 1" for that crop year.) In effect, this approach permits all calendar-related detail to be confined within INTMAT, exterior to the permanent software.

Figure 2—Weather data bank updating process

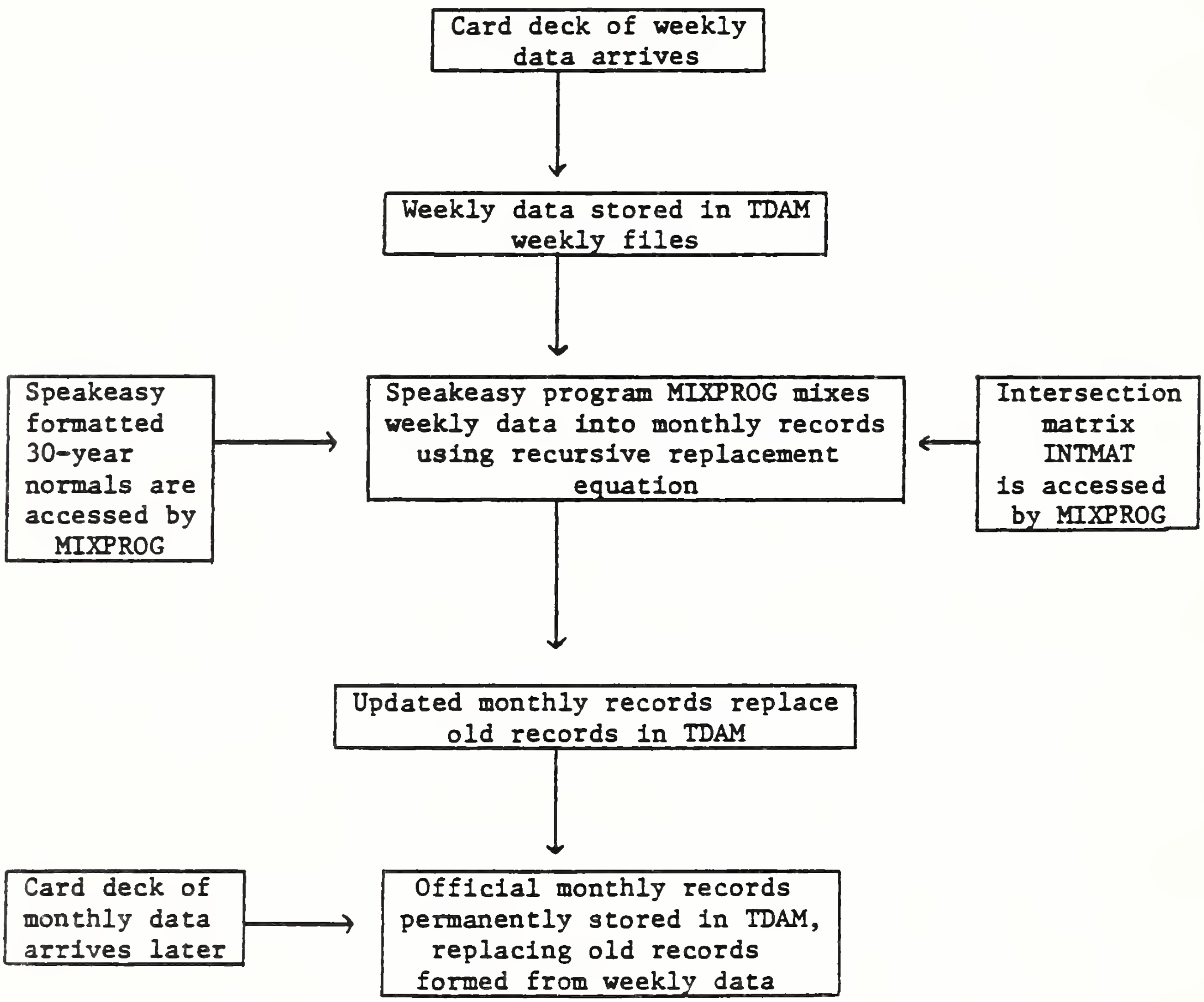


Figure 3--Recursive data-replacement equation

$$NEWVAL = OLDVAL + INTMAT(k,j) \cdot \left[\frac{WEEKVAL}{TIMETYP} - \frac{MONNORM}{MONLENGTH}\right]$$

This equation is used for each climatic division to mix temperature or precipitation data for week j into the provisional record for month k.

Variables:

NEWVAL is monthly record after mixing.
OLDVAL is monthly record before mixing.
INTMAT(k,j) is number of days common to month k and NOAA week j.
WEEKVAL is record for week j.
MONLENGTH is number of days in month k.
TIMETYP is 7 for precipitation, MONLENGTH for temperature.
MONNORM is 30-year monthly normal for climatic division (temperature or precipitation, as the case may be).

Since applied economic research not infrequently involves the consolidation of incommensurably-timed data-series, it might be mentioned that the approach described here--of capturing the essence of the mixing process in a fixed, abstract algorithm while confining all *ad hoc* time relationships within a simple, external "intersection matrix"--should be applicable in other contexts.

The Economic Data Bank

Need for an Economic Data Bank

Though weather is an important determinant of crop yields, it is by no means the only influence. Economic factors, in particular, play a measurable role. If fertilizer prices increase and farmers thus apply less fertilizer to their crops, yield may decrease. Or, if crop prices decrease and as a result farmers plant fewer acres, they may well remove from production the least-productive land first, so that yield (which, after all, is by definition the ratio of quantity-of-crop produced to land-area used) may increase.

Consistent with the responsibilities of ERS for economic research and analysis, it was anticipated from the beginning of the project that the regression equations ultimately developed would reflect the impact on yield not only of weather, but also of traditional economic factors such as crop acreages. However, as preliminary equations were developed, often in an atmosphere of urgency, the lack of a complete, accurate, and readily available set of state-level economic data proved a constant irritant. A good deal of data had to be copied hurriedly by hand from diverse statistical compendia and then typed into computer storage. The continuation of this situation was unacceptable, and it was determined that a definitive, computerized bank of state-level economic data would be necessary. Through much effort, by combining, correcting, and

extending data-series culled from various fragmentary sources, this bank was created and made an integral part of the developing system.

At its current stage of development, the economic data bank consists of annual records (1939 to present) of acres planted, acres harvested, and yield per harvested acre, for the U.S. and the 48 contiguous states. Corn, soybeans, sorghum, barley, oats, and several categories of wheat are represented.

Aside from their use in the creation of purely economic variables, the acreages data also play an indispensable role in the creation of agriculturally meaningful weather variables, for temperature and precipitation figures for a region are typically "weighted" by planted- or harvested acreages in order that weather effects be related fairly to their agricultural consequences.

Operation and Maintenance

The economic data, unlike the weather data, reside in (disk) files which are directly manipulated through the computer language Speakeasy (see [2]). Developed in the 1960's at the Argonne National Laboratories, Speakeasy was originally intended for atomic-energy research. However, its impressive capabilities in performing high-level mathematical, logical, and editorial operations, especially on large arrays of numbers or characters, prompted its adoption in other disciplines. The version maintained in ERS (Fedeasy) contains econometric routines added by the Federal Reserve Bank (see [5]).

The economic data bank is specifically designed to exploit the powerful capabilities of Speakeasy. Data are stored in matrix-like "boxes," one for each family of data, with rows corresponding to years and columns corresponding to states or U.S. Thus, for example, CRNBXAP ("corn box--acres planted") is an array of columns and rows containing all of the annual state and U.S. data on acres planted for corn. Each data box in its entirety, or portions of it, may be accessed and manipulated with unusual ease.

Figure 4 illustrates the operation of the data bank, in an interactive ("on-line") mode, by a user who wishes to obtain the historical annual ratios of harvested- to planted acreages for corn, for the combined region formed by Illinois, Iowa, Indiana, Ohio, and Nebraska. First, the corn acreage data is brought from storage into the "workspace." Then, in a single line, the states are selected, the regional totals of harvested and planted acreages formed, and the annual ratios computed. Finally, a printout of the results is generated. Observe that, within the data boxes, states are identified by simple U.S. Postal Service abbreviations.

An important feature of the data bank is the ease with which data may be updated. Though there are hundreds of time-series to be updated several times yearly, perhaps only two staff-days per year are required for data maintenance. Essentially, to update a data box, the operator merely types the new figures in the same order as they appear in the publication being used; the System already "knows" which states correspond. Then, a brief command serves to update all time-series in the data box simultaneously. This procedure is in marked contrast to what is required by many economic data banks, where one would have to enter a separate variable name for each of several hundred time-series being updated. Such repetitious entry of information other than the raw data is inefficient and logically unnecessary. It arises from organizing economic data as separate time-series rather than as structured (e.g., matrix-like) <u>collections</u> of (related) time-series.

The storing of data in a matrix-like format indeed conveys numerous benefits in a system of this type. For example, suppose that one wishes to have a simple technique for checking whether data have been (inadvertently) altered since the last update. If a data box is of dimensions m by n, one can multiply it coordinatewise by the m by n array whose elements are the integers 1,..., mn.

Figure 4--Example of economic-data-bank use*

```
:get crnbxah; get crnbxap
:ratio=sumrows(crnbxah(,(il ia in oh ne)))/sumrows(crnbxap(,(il ia in oh ne)))
:tabulate(year ratio)
```

YEAR	RATIO	YEAR	RATIO	YEAR	RATIO	YEAR	RATIO
.....	.....	.....	.....	.....	.....	.....	.....
1939	.90	1950	.95	1961	.96	1972	.94
1940	.89	1951	.94	1962	.95	1973	.95
1941	.95	1952	.96	1963	.95	1974	.92
1942	.95	1953	.95	1964	.95	1975	.94
1943	.93	1954	.94	1965	.95	1976	.94
1944	.94	1955	.91	1966	.95	1977	.94
1945	.93	1956	.88	1967	.94	1978	.95
1946	.95	1957	.95	1968	.94	1979	.96
1947	.92	1958	.95	1969	.94	1980	.95
1948	.96	1959	.96	1970	.94	1981	.96
1949	.96	1960	.96	1971	.95		

* In this and subsequent figures, user input is framed to distinguish it from computer output.

One can then sum all the mn elements of the "product box" to obtain a single number, which can be recorded. Later, whenever desired, the same procedure can be applied again to the data box. If the number which results is the same as obtained previously, then one can have a high degree of confidence that the data box has not changed. If the two numbers are not the same, then, by comparing the m row sums and n column sums of the "product box" computed at the two different times, noting which columns and rows have changed, and examining their intersections, one can infer which data-box elements have been altered. In this manner, the $m + n + 1$ numbers calculated from the data box can help safeguard the mn numbers stored in it.

The Data-Manipulation Subsystem

Need for the Subsystem. Examples of its use

Early experience in using the weather data bank demonstrated that a special data-access-and-manipulation system would have to be developed if the user were not to be overwhelmed by the sheer mass of data and number of operations required in ordinary applications.

The derivation of a typical corn-yield regression equation offers a case in point. A regional approach to corn-yield estimation might easily involve six states, fifty climatic divisions, and (since temperature and precipitation are both considered) one hundred weather time-series. The series would have to be weighted by published NOAA "area coefficients" to compensate for differences in area between climatic divisions of a state, and by further coefficients to account for differences in area between states. Once the series had been combined to form two regional aggregate series (one for temperature, one for precipitation), perhaps half-a-dozen different monthly subseries would have to be extracted from each for statistical testing in the desired regression.

In the initial phase of the project, the need for immediate, usable results made procedures of this explicit complexity unavoidable. However, as time permitted, a sequence of increasingly powerful data-manipulation routines was developed, culminating in the data-manipulation subsystem now in place.

The common language of this subsystem, and the key to its flexibility and sophistication, is the computer language Speakeasy described earlier. The heart of the subsystem is the Speakeasy program UTREV ("utility-revised"). UTREV is capable of manipulating weather records for any chosen groupings of years, months, states, and state climatic divisions. Its usefulness is most easily grasped through several actual examples of its operation as experienced interactively by the user at the computer terminal:

Example 1--As part of a study on corn yields, the user wishes to create annual time-series representing early-season (September-May) and late-season (June-August) precipitation for a large corn-production region.

He enters (Figure 5) the weather type ("PRCP"); the state names ("IA," "NE," etc.); for each state, the climatic divisions where corn is produced ("ALL;" "3 6 9;" etc.); the full list of months considered ("SEP," "OCT," etc.); and the starting and ending years ("1931," "1980"). He then iteratively selects two groupings of months for aggregation and assigns names to the resulting data arrays.

At this stage, the user has created separate annual early- and late-season series for each state. With a few brief Speakeasy commands (not shown), he can now combine these series into regional series, each of which represents the total early- or late-season precipitation for the entire region under consideration (Figure 6). Alternatively, with another command, he can weight the separate state series with state harvested- or planted acreages for corn, arriving

Figure 5--Using UTREV: Generating precipitation series for corn-production region

```
SPECIFY WEATHER TYPE (TEMP, PRCP, OR PDI)
WTYPE = prcp
ENTER STATE NAME ABBREVIATIONS
STATNAMS = ia ne mn il in oh wi mo sd mi
ENTER "ALL" OR LIST CHOSEN CLIMATIC SUBDIVISIONS IN
ORDER WITHOUT LEADING ZEROES
SUBDVSIA = all
SUBDVSNE = 3 6 9
SUBDVSMN = 4 5 6 7 8 9
SUBDVSIL = all
SUBDVSIN = all
SUBDVSOH = all
SUBDVSWI = 4 5 6 7 8 9
SUBDVSMO = 1 2 3 4 6
SUBDVSSD = 2 3 6 7 8 9
SUBDVSMI = 5 6 7 8 9 10
ENTER MONTHS IN ORDER
MONVALS = sep oct nov dec jan feb mar apr may jun jul aug
ENTER STARTING AND ENDING YEARS FOR FIRST MONTH
STARTYR = 1931
LASTYR = 1980
WANT TO COMBINE MONTHS?
ANS = yes
ENTER MONTHS TO BE COMBINED IN ORDER
CHOICES = sep oct nov dec jan feb mar apr may
WITHOUT BLANKS, ENTER NAME OF FORTHCOMING ACCUMULATION ARRAY early
EARLY IS A 50 BY 10 REAL ARRAY
WANT TO COMBINE MONTHS AGAIN?
ANS = yes
ENTER MONTHS TO BE COMBINED IN ORDER
CHOICES = jun jul aug
WITHOUT BLANKS, ENTER NAME OF FORTHCOMING ACCUMULATION ARRAY late
LATE IS A 50 BY 10 REAL ARRAY
WANT TO COMBINE MONTHS AGAIN?
ANS = no
:_
```

Figure 6--Precipitation series for corn-production region (unweighted)

CROPYEAR	EARLYREG	LATEREG	CROPYEAR	EARLYREG	LATEREG
1932	23.88	11.70	1957	19.61	12.46
1933	23.14	7.24	1958	18.91	13.88
1934	13.69	8.80	1959	20.70	9.73
1935	24.84	11.85	1960	23.87	11.29
1936	17.18	5.90	1961	19.77	10.74
1937	24.86	10.61	1962	24.34	11.36
1938	22.62	11.37	1963	17.67	10.40
1939	20.41	11.93	1964	18.08	10.51
1940	15.62	11.22	1965	20.98	11.48
1941	17.03	9.99	1966	22.38	9.95
1942	26.77	12.53	1967	19.82	11.03
1943	23.43	12.00	1968	21.48	11.58
1944	21.10	11.91	1969	23.27	12.55
1945	22.59	11.49	1970	21.30	9.83
1946	22.24	10.40	1971	22.23	9.42
1947	24.03	11.47	1972	22.67	11.41
1948	20.93	11.03	1973	29.07	10.45
1949	21.56	11.35	1974	26.88	10.21
1950	24.53	11.81	1975	22.21	11.84
1951	22.12	14.03	1976	20.56	7.64
1952	22.98	11.55	1977	16.94	12.92
1953	18.00	9.78	1978	24.23	11.91
1954	16.55	11.40	1979	22.95	13.22
1955	20.34	9.87	1980	17.31	12.83
1956	18.31	11.45	1981	18.83	14.52

at series which represent the total early- or late-season precipitation for the farmland actually utilized for corn production (Figure 7; compare Figure 6).

Example 2--In order to investigate the relationship between low temperatures and cattle death losses, the user wishes to obtain annual time-series for January and February average temperatures over a major livestock-producing region. He enters the weather type, the states, the respective climatic divisions, and the months and years (Figure 8) and calls for separate January and February (state) series to be formed (Figure 9 displays these separate series for February). These series are then aggregated to form the regional series desired (Figure 10). Alternatively, the state series could be weighted by state herd sizes, if desired.

The next example highlights one of the most useful features of the data-manipulation subsystem--the "probability operator."

Example 3--Concerned about the prospects for the winter wheat crop in the Texas Panhandle, the user wishes to determine the probabilities (based on historical frequencies) of various weather events for the month of April (such as the probability of a "hot and dry" April).

As before, he enters the appropriate information (first for precipitation, then for temperature), choosing Texas climatic divisions 1 and 2 to represent the Panhandle (Figure 11). He thereby obtains annual time-series for April precipitation and temperature for the Panhandle (Figure 12).

To obtain the desired probabilities, he defines "HOT" as the event that the April temperature over the Panhandle is greater than ("GT") average and defines "DRY" analogously (Figure 13). (Other definitions would of course be possible.) He now merely enters his probability questions in a natural form as

Figure 7--Precipitation series for corn-production region (weighted by area planted)

CROPYEAR	EARLYWT	LATEWT	CROPYEAR	EARLYWT	LATEWT
1939	20.16	12.03	1961	19.16	10.84
1940	15.06	11.24	1962	24.73	11.88
1941	17.02	9.98	1963	17.22	10.85
1942	26.63	12.63	1964	18.00	10.76
1943	22.27	12.36	1965	21.47	11.72
1944	21.27	12.54	1966	22.83	9.88
1945	22.10	11.63	1967	19.65	11.53
1946	21.59	10.71	1968	21.25	11.69
1947	23.79	11.70	1969	23.59	13.10
1948	20.59	11.07	1970	21.27	10.11
1949	21.55	11.52	1971	22.40	9.57
1950	23.85	12.20	1972	22.68	11.96
1951	22.24	14.73	1973	29.09	10.73
1952	22.61	12.14	1974	26.98	10.39
1953	17.64	9.83	1975	21.89	11.90
1954	16.61	11.93	1976	20.66	7.66
1955	19.54	9.70	1977	16.84	13.48
1956	17.49	11.38	1978	23.86	12.09
1957	19.12	12.90	1979	23.16	13.56
1958	18.54	14.16	1980	17.26	13.34
1959	20.94	10.03	1981	18.31	14.61
1960	23.94	11.73			

:_

Figure 8--Using UTREV in cattle death-loss study

```
SPECIFY WEATHER TYPE (TEMP, PRCP, OR PDI)
WTYPE = temp
ENTER STATE NAME ABBREVIATIONS
STATNAMS = mt wy nd sd
ENTER "ALL" OR LIST CHOSEN CLIMATIC SUBDIVISIONS IN
ORDER WITHOUT LEADING ZEROES
SUBDVSMT = 3 4 5 6 7
SUBDVSWY = 2 4 5 6 7 8
SUBDVSND = 1 4 7 8
SUBDVSSD = 1 4 5
ENTER MONTHS IN ORDER
MONVALS = jan feb
ENTER STARTING AND ENDING YEARS FOR FIRST MONTH
STARTYR = 1931
LASTYR = 1981
WANT TO COMBINE MONTHS?
ANS = yes
ENTER MONTHS TO BE COMBINED IN ORDER
CHOICES = jan
WITHOUT BLANKS, ENTER NAME OF FORTHCOMING ACCUMULATION ARRAY jantemp
JANTEMP IS A 51 BY 4 REAL ARRAY
WANT TO COMBINE MONTHS AGAIN?
ANS = yes
ENTER MONTHS TO BE COMBINED IN ORDER
CHOICES = feb
WITHOUT BLANKS, ENTER NAME OF FORTHCOMING ACCUMULATION ARRAY febtemp
FEBTEMP IS A 51 BY 4 REAL ARRAY
WANT TO COMBINE MONTHS AGAIN?
ANS = no
```

Figure 9--February temperature series by state (using selected climatic divisions)

FEBTEMP

YEAR	MT	WY	ND	SD	YEAR	MT	WY	ND	SD
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
1931	33.2	30.9	29.9	33.9	1957	20.1	28.9	14.4	25.4
1932	22.8	26.3	15.2	24.1	1958	21.4	28.3	12.6	21.0
1933	15.1	16.1	9.3	18.2	1959	12.9	22.0	6.5	17.2
1934	30.3	30.5	23.1	29.9	1960	19.0	20.5	12.6	18.1
1935	29.6	30.0	25.6	32.7	1961	31.4	31.2	20.8	30.2
1936	6.5	9.1	-12.8	-1.1	1962	19.9	25.0	10.1	22.1
1937	14.4	21.7	8.7	19.8	1963	30.2	32.3	17.5	28.8
1938	15.3	26.1	7.8	22.6	1964	28.5	22.4	22.9	27.0
1939	12.3	15.2	1.6	13.6	1965	20.4	23.2	10.8	22.6
1940	21.0	26.2	15.7	23.0	1966	19.3	23.0	10.7	17.3
1941	25.4	28.9	15.8	25.8	1967	26.1	27.6	11.9	24.7
1942	18.9	19.4	16.0	21.9	1968	26.3	29.0	14.5	25.2
1943	26.0	29.8	18.0	30.3	1969	17.2	26.1	13.6	22.1
1944	21.5	22.9	12.8	18.8	1970	27.1	31.1	16.7	29.2
1945	22.3	25.2	17.1	23.6	1971	23.0	24.1	14.5	20.7
1946	25.0	27.9	14.5	29.5	1972	19.4	27.8	8.7	22.0
1947	18.9	23.2	10.5	19.2	1973	26.2	26.2	20.8	27.6
1948	17.5	21.2	9.4	19.2	1974	29.9	29.6	18.7	30.7
1949	11.2	19.4	3.6	14.8	1975	14.9	20.5	11.5	16.6
1950	24.6	30.0	11.5	24.8	1976	30.3	29.4	25.9	33.3
1951	20.3	28.9	12.8	26.7	1977	32.9	29.8	24.5	32.3
1952	22.9	26.3	18.2	25.6	1978	14.7	19.7	7.2	13.3
1953	27.5	26.0	20.9	25.7	1979	12.8	22.0	1.6	13.1
1954	35.9	36.2	33.0	39.5	1980	24.3	26.5	15.6	25.0
1955	17.2	20.0	9.1	18.0	1981	27.8	27.8	21.4	27.4
1956	17.7	21.4	11.2	21.0					

:_

Figure 10--January-February temperature series for livestock-producing region

YEAR	TJANREG	TFEBREG	YEAR	TJANREG	TFEBREG	YEAR	TJANREG	TFEBREG
.....	....	....	.....	....	....	.....	....	....
1931	27.6	32.2	1948	21.7	17.3	1965	18.9	19.8
1932	15.4	22.5	1949	5.3	12.3	1966	11.1	18.6
1933	21.6	14.8	1950	.6	23.8	1967	21.6	24.0
1934	27.1	29.2	1951	14.1	21.9	1968	17.0	24.9
1935	16.9	29.4	1952	13.4	23.3	1969	9.7	19.3
1936	12.0	3.2	1953	27.0	25.9	1970	14.2	26.7
1937	-.3	15.8	1954	12.2	35.9	1971	13.7	21.6
1938	22.6	17.5	1955	19.7	16.7	1972	11.1	20.0
1939	25.9	11.4	1956	16.3	17.9	1973	20.4	25.5
1940	10.8	21.6	1957	7.4	21.9	1974	16.5	28.2
1941	22.1	24.7	1958	27.5	21.6	1975	20.1	15.9
1942	21.2	18.9	1959	16.0	14.5	1976	20.2	29.8
1943	10.0	26.1	1960	17.6	18.2	1977	11.4	30.8
1944	25.1	20.2	1961	24.1	29.6	1978	8.0	14.5
1945	21.6	22.3	1962	15.6	19.8	1979	3.2	13.2
1946	23.5	24.6	1963	9.4	28.5	1980	14.4	23.5
1947	21.8	18.6	1964	23.1	26.0	1981	28.7	26.8

:_

Figure 11 --Using UTREV in Texas winter wheat study

```
SPECIFY WEATHER TYPE (TEMP, PRCP, OR PDI)
WTYPE = [prcp]
ENTER STATE NAME ABBREVIATIONS
STATNAMS = [tx]
ENTER "ALL" OR LIST CHOSEN CLIMATIC SUBDIVISIONS IN
ORDER WITHOUT LEADING ZEROES
SUBDVSTX = [1 2]
ENTER MONTHS IN ORDER
MONVALS = [apr]
ENTER STARTING AND ENDING YEARS FOR FIRST MONTH
STARTYR = [1931]
LASTYR = [1981]
WANT TO COMBINE MONTHS?
ANS = [no]
:_

SPECIFY WEATHER TYPE (TEMP, PRCP, OR PDI)
WTYPE = [temp]
ENTER STATE NAME ABBREVIATIONS
STATNAMS = [tx]
ENTER "ALL" OR LIST CHOSEN CLIMATIC SUBDIVISIONS IN
ORDER WITHOUT LEADING ZEROES
SUBDVSTX = [1 2]
ENTER MONTHS IN ORDER
MONVALS = [apr]
ENTER STARTING AND ENDING YEARS FOR FIRST MONTH
STARTYR = [1931]
LASTYR = [1981]
WANT TO COMBINE MONTHS?
ANS = [no]
:_
```

Figure 12--Texas Panhandle precipitation and temperature series

YEAR	PANPRCP	PANTEMP	YEAR	PANPRCP	PANTEMP	YEAR	PANPRCP	PANTEMP
.....	....	....	.....	....	....	.....	....	....
1931	2.25	56.1	1948	.65	65.4	1965	1.24	63.7
1932	2.42	61.9	1949	2.18	57.5	1966	2.58	58.9
1933	.70	58.7	1950	1.39	60.3	1967	1.52	65.1
1934	1.40	62.3	1951	.84	58.6	1968	1.27	57.6
1935	.72	60.3	1952	2.58	58.7	1969	1.23	61.3
1936	1.08	60.1	1953	1.08	59.1	1970	1.29	58.6
1937	.98	60.6	1954	2.44	65.3	1971	1.13	59.8
1938	1.44	59.0	1955	.64	63.3	1972	.74	64.4
1939	1.30	60.9	1956	.56	59.6	1973	2.44	53.8
1940	1.84	59.4	1957	4.14	57.3	1974	1.17	62.0
1941	3.21	59.1	1958	2.20	56.3	1975	1.06	57.9
1942	4.48	60.5	1959	1.40	59.4	1976	3.05	61.1
1943	1.39	64.5	1960	.76	63.1	1977	3.33	59.5
1944	1.37	57.7	1961	.23	59.9	1978	.45	64.6
1945	1.62	56.1	1962	1.67	60.5	1979	1.29	59.5
1946	.84	65.5	1963	1.02	65.7	1980	1.22	57.8
1947	1.93	57.6	1964	.42	62.5	1981	2.23	63.9

Figure 13—Application of the probability operator

```
: hot=pantemp.gt.mean(pantemp)
:
: dry=panprcp.lt.mean(panprcp)
:
: prob(hot.and.dry)
PROB(HOT.AND.DRY) =   .3
:
: prob(hot.or.dry)
PROB(HOT.OR.DRY) =   .8
:
: prob(hot)
PROB(HOT) =   .4
:
: prob(dry)
PROB(DRY) =   .6
:
: prob(panprcp.gt.2)
PROB(PANPRCP.GT.2) =   .3
:_
```

shown, and the System responds with the answers (in this case, concluding with the probability of obtaining more than two inches of precipitation).

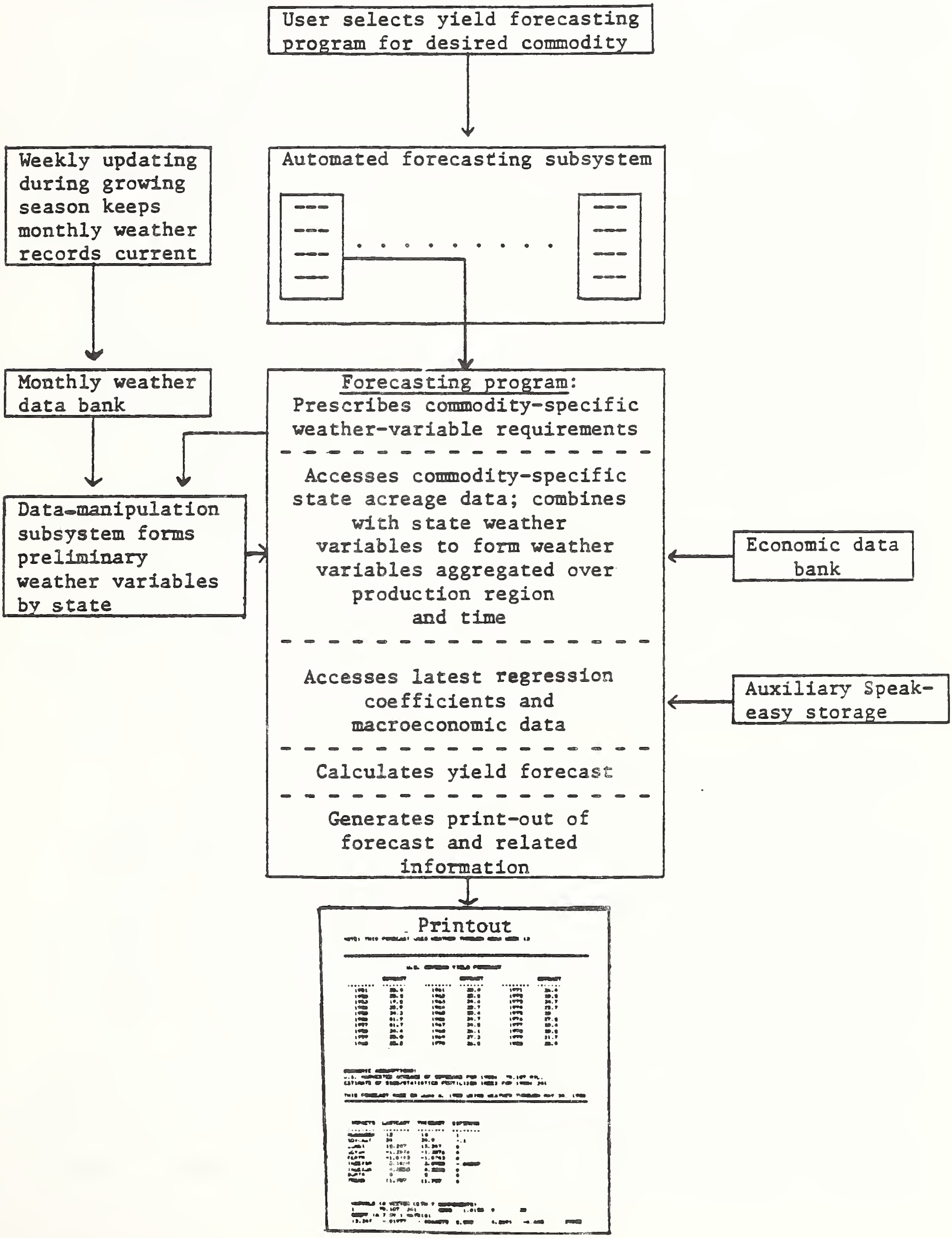
In addition to UTREV, the data-manipulation subsystem contains a variety of other specialized programs and auxiliary data sets. Notable among the latter are collections of state areas, climatic-division area coefficients, and 30-year monthly weather normals (by climatic division). Furthermore, operation of the subsystem implies immediate access to the full range of standard Speakeasy and Fedeasy operations, permitting extensive statistical and mathematical analyses to be conducted, interactively, at any stage of a study.

The Automated Forecasting Subsystem

The automated yield-forecasting subsystem draws upon all the development efforts previously described. This subsystem enables the user to generate crop-yield forecasts which reflect the impact of weather up to approximately the previous week. By comparing his forecasts from week to week, the user can "track" the effect which weather is having on expected crop yields.

The subsystem consists of Speakeasy programs which generate U.S. crop-yield forecasts for corn, soybeans, sorghum, and barley, using regression equations which reflect both traditional economic factors and "weather-up-to-the-present." To generate a forecast, the user merely gets into the interactive Speakeasy mode on the computer and enters the name of the forecasting program to be used. Using modified versions of UTREV, each program automatically selects weather records (appropriate for that crop) from the weather data bank, weights them with acreage data from the economic data bank to form aggregate weather variables, and applies previously-determined regression coefficients to these and selected macroeconomic variables to produce the crop-yield forecast, which is then printed out. Figure 14 schematizes the

Figure 14--Operation of automated forecasting subsystem



operation of the automated forecasting subsystem in its relationship to the entire Automated Weather/Yield System.

Figure 15 outlines a soybean-yield equation whose forecasts have been automated. The underlying regression model treats expected U.S. yield for soybeans as being linearly related to a constant term, the harvested acreage of soybeans (estimated prior to harvest as a fraction of planted acreage), an index of fertilizer prices, weather variables representing early- and late-season precipitation, a "dummy variable" used to represent the effects of a peculiar weather pattern in 1974, and a linear time-trend variable used as a proxy for technological progress. (It should be noted that, although the expected yield for an individual soybean plant might appear to be more properly modelled as a quadratic, rather than linear, function of precipitation, a <u>set</u> of plants occupying a geographical region is quite another entity from a probabilistic standpoint. An extensive discussion of this point, however, is outside the scope of the present work). The t statistics shown in Figure 15 for the precipitation variables INDEXSM and INDEXJA--especially for the latter--attest to the feasibility of using aggregate weather variables in models of crop yield.

Figure 16 shows the printout of a soybean-yield forecast which has been generated by the automated forecasting subsystem. The actual forecast, 28.9 bushels per acre, appears next to the (then) current year, 1980. The fitted values for 1951-79 are included for convenient comparison. The printout displays the economic assumptions used, the date of the forecast, and the date (and NOAA week) up to which the weather data bank has been updated. Also displayed are the values of the regression variables (VARVALS) and the corresponding regression coefficients (COEFFS). The last seven lines of the columns LASTCAST and THISCAST show the impact of each model variable (that is, the

Figure 15--U.S. soybean-yield regression results

Variable	Estimated coefficient	Standard error	T statistic
b_o (constant)	13.20	1.69	7.85
SOYSH	-0.02	0.06	-0.31
FERTM	-0.004	0.006	-0.74
INDEXSM	2.56	1.44	1.78
INDEXJA	4.27	0.83	5.13
DUM74	-4.44	0.97	-4.57
TREND	0.39	0.11	3.58

R squared = 0.95

R squared (corrected) = 0.94

Durbin-Watson statistic = 2.24

Standard error = 0.82

Data from years 1951-79

Regression Variables:

SOYSH is the U.S. harvested acreage of soybeans in millions of acres.

FERTM is a fertilizer price index.

INDEXSM is an early-season precipitation index.

INDEXJA is a late-season precipitation index.

DUM74 is a dummy variable equaling 1 in 1974 and 0 in other years.

TREND is a linear time trend.

Figure 16--A forecast from the automated forecasting subsystem

NOTE: THIS FORECAST USES WEATHER THROUGH NOAA WEEK 13

****

U.S. SOYBEAN YIELD FORECAST

....	SOYCAST	....	SOYCAST	....	SOYCAST ....
1951	20.9	1961	23.9	1971	26.4
1952	20.3	1962	25.2	1972	28.5
1953	19.2	1963	24.4	1973	28.7
1954	20.9	1964	23.7	1974	23.7
1955	20.3	1965	25.4	1975	28
1956	21.9	1966	24.7	1976	27.2
1957	21.7	1967	24.5	1977	30.4
1958	24.4	1968	26.1	1978	30.2
1959	22.8	1969	27.2	1979	31.7
1960	23.2	1970	26.5	1980	28.9

ECONOMIC ASSUMPTIONS:
U.S. HARVESTED ACREAGE OF SOYBEANS FOR 1980: 70.187 MIL.
ESTIMATE OF ESCS/STATISTICS FERTILIZER INDEX FOR 1980: 241

THIS FORECAST MADE ON June 6, 1980 USING WEATHER THROUGH MAY 24, 1980
****

IMPACTS	LASTCAST	THISCAST	DIF2MIN1
....	....	....	....
NOAAWEEK	12	13	1
SOYCAST	29	28.9	-.1
CONST	13.247	13.247	0
SOYSH	-1.3876	-1.3876	0
FERTM	-1.0743	-1.0743	0
INDEXSM	2.1469	2.0988	-.04809
INDEXJA	4.3353	4.3353	0
DUM74	0	0	0
TREND	11.707	11.707	0

VARVALS (A VECTOR WITH 7 COMPONENTS)
1 70.187 241 .8205 1.0155 0 30
COEFF (A 7 BY 1 MATRIX)
13.247 -.01977 -.0044575 2.558 4.2691 -4.445 .39025

value of the variable times its regression coefficient) on the last forecast ("LASTCAST") and on the current forecast ("THISCAST"), while the column DIF2MIN1 shows the difference in these impacts. Thus, this particular printout reveals that, when the weather data bank was updated from NOAA Week 12 by the inclusion of weather data for NOAA Week 13 (ending May 24, 1980), the expected U.S. soybean yield dropped by 0.1 bushels per acre, and this drop (with allowance for some discrepancy due to rounding) was attributable to a drop of approximately 0.048 in the impact of early-season (September-May) precipitation below its previously expected level.

Summary

The Automated Weather/Yield System has developed into an information, research, and forecasting tool capable of responding to a variety of needs with only minimal demands made of the user. The weather data bank itself has already performed an auxiliary role as the key data source for portions of a major USDA study [10], and, in June, 1982, it was formally selected for retention by the National Archives. The overall System is called upon regularly to provide information for studies in government and academia encompassing such diverse fields of application as the economic effects of drought, pesticide research, energy-use in agriculture, disaster-area relief, crop-yield research, agricultural productivity, and seasonality of consumer demand.

Other applications and extensions can be envisioned, ranging from improved estimation of pig death losses in econometric models to better understanding of the economic impacts of weather on rural populations. Of particular significance could be the broadening of the system to include foreign countries. This would provide a foundation for strengthening yield-modelling efforts in the context of the world agricultural economy.

CHAPTER 2: A TECHNICAL VIEW

This chapter provides a highly technical discussion of the design and operation of the Automated Weather/Yield System. Its aim is to provide the interested general reader, the system operator, or persons wishing to install similar systems at their institutions, with a thorough and precise technical understanding of the System in all its details. The chapter parallels Chapter 1 in that it treats, in turn, the weather data bank, economic data bank, data-manipulation subsystem, and automated forecasting subsystem. The reader is assumed to be fluent in the computer language Speakeasy (see [2]). The Fedeasy dialect is recommended (see [6]).

Chapter 2 is written in three languages--English, Speakeasy, and (cf. p.54) mathematics. There should be no confusion between the latter two. However, because quotation marks are part of both English and Speakeasy, yet have different uses in each, an instruction to the reader such as

type "UTREV"

would be ambiguous. To avoid such ambiguities, the author has chosen to omit certain quotation marks in this chapter which ordinarily would be prescribed by proper English usage.

The Weather Data Bank

General Description

The weather data bank contains three types of weather (more accurately, climatic) data--temperature (in degrees Fahrenheit), precipitation (in inches), and the Palmer Drought Index. The permanently-stored data are all monthly; however, some weekly data are maintained temporarily for use in the updating process (cf. pp.33-5).

For each of the NOAA climatic divisions (cf. p.2 and Figure 1), the weather data bank contains three monthly time-series--one for the average temperature, one for the total precipitation, and one for the Palmer Drought Index, over that climatic division. The series for temperature and precipitation extend from January, 1931 to the current month (and beyond, as will be explained). The series for the Palmer Drought Index are mostly complete through 1975. (Because the Palmer series have received only infrequent use, it is recommended that, as an economy measure, they be considered for archival, but in a manner which will insure their continued availability. One problem with the use of the Palmer Drought Indices has been that they are climatic-division-specific, in the sense that the Index measures drought within a climatic division as a deviation from the division's "normal." Thus, a desert and a rain forest, if they are each experiencing "normal" conditions, might receive the same Index value of 0. Accordingly, Palmer Drought Indices for two-or-more climatic divisions cannot meaningfully be aggregated.)

The monthly weather time-series described above are stored in logical group WEAM in TDAM (see p.2). Their naming scheme is as follows: Precipitation-variable names have nine characters, namely the five characters PRCP1, followed by a standard two-character U.S. Postal Service state-name abbreviation (see Figure 17), followed by two digits giving the climatic-division number (here, 01, 02,...,10 correspond to 1, 2,..., 10). (Note that the divisions are numbered separately within each state. No state has more than ten divisions. The reader is cautioned that there exists no "division 4" of Nebraska, due apparently to an oversight in NOAA's original numbering of the climatic divisions. Nebraska has divisions numbered 1, 2, 3, 5, 6, 7, 8, 9; these do, however, comprise the entire state. All other states have their divisions numbered consecutively, beginning with 1.) The temperature series and the Palmer Drought Index series have names of nine and eight characters, respectively, the former beginning TEMP1, the

Figure 17--State-name abbreviations

1.	AL	Alabama
2.	AZ	Arizona
3.	AR	Arkansas
4.	CA	California
5.	CO	Colorado
6.	CT	Connecticut
7.	DE	Delaware
8.	FL	Florida
9.	GA	Georgia
10.	ID	Idaho
11.	IL	Illinois
12.	IN	Indiana
13.	IA	Iowa
14.	KS	Kansas
15.	KY	Kentucky
16.	LA	Louisiana
17.	ME	Maine
18.	MD	Maryland
19.	MA	Massachusetts
20.	MI	Michigan
21.	MN	Minnesota
22.	MS	Mississippi
23.	MO	Missouri
24.	MT	Montana
25.	NE	Nebraska
26.	NV	Nevada
27.	NH	New Hampshire
28.	NJ	New Jersey
29.	NM	New Mexico
30.	NY	New York
31.	NC	North Carolina
32.	ND	North Dakota
33.	OH	Ohio
34.	OK	Oklahoma
35.	OR	Oregon
36.	PA	Pennsylvania
37.	RI	Rhode Island
38.	SC	South Carolina
39.	SD	South Dakota
40.	TN	Tennessee
41.	TX	Texas
42.	UT	Utah
43.	VT	Vermont
44.	VA	Virginia
45.	WA	Washington
46.	WV	West Virginia
47.	WI	Wisconsin
48.	WY	Wyoming

latter PDI1. The final four characters of the names are formed exactly as for the precipitation-series names. Thus, typical examples of weather-series names are PRCP1IL04, TEMP1KS02, and PDI1OH10.

For completeness, it may be mentioned that the weather data bank currently also contains (in Speakeasy library storage) complete monthly records (1931-80) of temperature and precipitation for each of the forty-eight contiguous states and each of the ten U.S. farm production regions. However, as these records were developed in connection with the preparation of [11] and not intended to be an "official" part of the weather data bank, a detailed description is not given here. Details may be obtained from the author.

Updating the Temperature and Precipitation Series

As an aid in forecasting, the WEAM temperature and precipitation series always extend through at least the full year following the current one. For each future month, the time-series element is taken to be the thirty-year normal (i.e., average; based on 1941-70) of the temperature or precipitation for the corresponding climatic division and month. (NOAA typically uses thirty-year periods for computing normals, apparently in the belief that such a period is long enough to reveal statistical patterns but short enough to avoid structural changes in the underlying climate.) Computer card decks containing NOAA's published monthly temperature and precipitation data are received from the National Climatic Center approximately six weeks after the end of a month, and these data are loaded into the data bank by the ERS Data Services Center (DSC), thereby redefining various elements of the stored time-series. Furthermore, during the crop-growing season, NOAA publishes provisional weekly values of its climatic-division weather data. These data are received by ERS (also in card-deck form) a few days after the end of a week and are loaded by DSC into the nominally annual TDAM logical group WEAA (the annual frequency being used

because TDAM does not provide "officially" for storage of weekly data). The data are stored in separate weekly time-series whose names are exactly the same as those of the monthly time-series, with the exception that the fifth character from the end of each name is "W" instead of "1" (hence: PRCPWIL03 instead of PRCP1IL03). The weekly time-series are retained only temporarily for use in the mixing process (to be explained shortly). They are each fifty-two elements long and employ locations 1901001 through 1952001 (corresponding to NOAA's week numbers 1, 2, 3,... for that crop season). Prior to arrival of a week's data, its time-series element is set equal to the TDAM default value of -999999.

As each new set of weekly data arrives and is stored, it is "mixed" into the monthly data to which that week applies, replacing one-or-more days of "normal" weather with the same number of days of estimated "actual" weather. (For example, if the week is contained entirely within a single month, then seven days of normal precipitation are replaced by seven days of "actual" precipitation, each of the latter being estimated as one-seventh of the week's total precipitation for that division. If the month has thirty-one days, then 7/31 of the monthly normal is taken to represent the seven days of normal precipitation.) Thus, for months during the crop-growing season, the monthly temperature and precipitation series originally contain normals, then successively more "real" mixtures of "actual" data and normals (reaching a temporary plateau as a mixture solely of "actual" weekly data), then, ultimately (when the "final" monthly data are received), the definitive monthly values.

The Speakeasy program MIXPROG was created to accomplish the weekly mixing process. However, due to the unusually high cost of accessing and revising TDAM variables, this program is extremely expensive to run. If the weather data is transferred to Speakeasy MYKEEP storage, a revised version of MIXPROG should

be quite inexpensive to run. Recently, software designed for the author by DSC has been used to accomplish the weekly mixing process. Copies of DSC's documentation of this software are available from the author.

As a technical convenience, there is a variable WKMXFLG ("weekly mixing flag") in WEAA (extending from 1901001 to 1952001, and with default value -999999) whose elements are changed by the DSC software to the corresponding week number when a set of weekly data has been mixed into the monthly data. (Thus, when the data for week 23 have been mixed, a 23 will be placed in location 1923001 of WKMXFLG.) Similarly, for the monthly time-series, there is a variable MNMXFLG ("monthly mixing flag") in WEAM (of the same length as the other monthly variables--with one element for each month and default value -999999) whose elements are changed to the number of the latest week whose data have been mixed into that month's data. Ultimately, each element in MNMXFLG is set equal to 100 when the definitive monthly NOAA data for that month have been received and loaded.

The Economic Data Bank

General Description. Structure and Manipulation of the Data Boxes

The economic data bank contains annual state and U.S. data on acres planted, acres harvested, and yield for corn, soybeans, sorghum, barley, oats, all wheat, winter wheat, durum wheat, and other spring wheat. The units are "1,000 acres" for the area planted- and area harvested data and "bushels per (harvested) acre" for the yield data.

In the construction of the economic data bank, great care was taken to insure the completeness and accuracy of the data (specifically, its precise agreement with the corresponding SRS publications (including latest corrections and revisions) as of January, 1982). In short, the data bank is designed to be definitive.

Each of the data bank's twenty-seven categories of data is stored in its own two-dimensional array (referred to in this document as a "data box") whose rows correspond to the years 1939-current year (row 1: 1939; row 2: 1940;...; row 43: 1981) and whose first forty-eight columns correspond to the forty-eight contiguous U.S. states in alphabetical order (column 1: Alabama; column 2: Arizona; ...; column 48: Wyoming). The data boxes are stored in MYKEEP. Each has fifty-one columns. Column 49 holds U.S.-level data, while column 50 displays the years corresponding to the rows. (Hence, for the data box for all corn, area planted, the entry in row 1, column 49 is the total U.S. area planted for all corn for 1939, while the entry in row 1, column 50 is the number 1939.) Column 51 is an auxiliary column for use as the System operator sees fit. Typically, this column would be used to record the date of the latest publication (say, 115.1982 to denote January 15, 1982) by means of which that row has been updated.

The System uses the following standard abbreviations for its nine crops:

CRN: corn

SOY: soybeans

SRG: sorghum

BAR: barley

OAT: oats

WTA: wheat all

WTW: wheat winter

WTD: wheat durum

WTO: wheat other (spring)

The standard abbreviations used for area planted, area harvested, and yield are:

AP

AH

YD

Standard two-character U.S. Postal Service state-name abbreviations (Figure 17) are used to refer to the forty-eight contiguous states (with US used for U.S.). Typically, states are organized in alphabetical order of (full) state names, followed by U.S. The System contains a 49 by 2 character array, STATELST ("state list"), whose rows display these abbreviated names. (Note that the rows of STATELST, though arising from an alphabetical ordering of full state names, are not themselves--as character strings--ordered alphabetically).

Figure 18 displays the names of the twenty-seven data boxes and provides precise references to the crop data which the boxes contain. (Note that "BX" connotes "box.") It will be advantageous for the operator, whenever possible, to treat the data boxes in the order given in Figure 18, or (if only certain of the data boxes are to be used) in a sub-order thereof. Toward this end, the System contains a 27 by 7 character array, BOXLIST, whose rows display the data-box names given in Figure 18, in the same order. Used in a loop in combination with OBJECT (as in OBJECT(BOXLIST(J))), BOXLIST provides a ready means of access to all twenty-seven data boxes. Note that, to access, say, just the nine acres-planted boxes, one could use BOXLIST(INTS(1 27 3)).

In order that the operator not need to concern himself with what column of a data box corresponds to what state, the System contains a program, GIVENUMS, which the operator may run prior to working with the data boxes. GIVENUMS merely defines Speakeasy scalars AL = 1, AZ = 2, etc., through

Figure 18--Data-box names and corresponding crop references

1.	CRNBXAP	All corn, area planted
2.	CRNBXAH	Corn for grain, area harvested
3.	CRNBXYD	Corn for grain, yield
4.	SOYBXAP	Soybeans, area planted
5.	SOYBXAH	Soybeans for beans, area harvested
6.	SOYBXYD	Soybeans for beans, yield
7.	SRGBXAP	All sorghum, area planted
8.	SRGBXAH	Sorghum for grain, area harvested
9.	SRGBXYD	Sorghum for grain, yield
10.	BARBXAP	Barley, area planted
11.	BARBXAH	Barley, area harvested
12.	BARBXYD	Barley, yield
13.	OATBXAP	Oats, area planted
14.	OATBXAH	Oats, area harvested
15.	OATBXYD	Oats, yield
16.	WTABXAP	All wheat, area planted
17.	WTABXAH	All wheat, area harvested
18.	WTABXYD	All wheat, yield
19.	WTWBXAP	Winter wheat, area planted
20.	WTWBXAH	Winter wheat, area harvested
21.	WTWBXYD	Winter wheat, yield
22.	WTDBXAP	Durum wheat, area planted
23.	WTDBXAH	Durum wheat, area harvested
24.	WTDBXYD	Durum wheat, yield
25.	WTOBXAP	Other spring wheat, area planted
26.	WTOBXAH	Other spring wheat, area harvested
27.	WTOBXYD	Other spring wheat, yield

WY = 48, US = 49. Thus, GIVENUMS sets each abbreviated state name equal to the data-box column number (n.b.: the STATELST row number) corresponding to that state. Hence, the operator may extract, say, the data for Illinois, Kansas, and the U.S. from CRNBXAP merely by entering CRNBXAP(,(IL KS US)). He need never concern himself with the column numbers themselves.

Similarly, the keptlisted namelist ROWYEARS, when retrieved with KEPTLIST(ROWYEARS), defines (at the time of this writing) scalars Y1939 = 1,..., Y1981 = 43, allowing reference to the data row for the year 1940, say, by CRNBXAP(Y1940). The operator need not concern himself with what row corresponds to what year. (Note that ROWYEARS could also have been constructed as a program fulfilling the same function, while GIVENUMS could have been constructed as a namelist.)

The "default" value for elements of the data boxes is -8888. A location (i.e., row/column intersection) in any of columns 1 through 49 holds the value -8888 if and only if the definitive SRS publication referring to that data type has not yet appeared, or has a blank, or no entry, corresponding to the state (or U.S.) and year which that location represents. Column 50 contains no -8888's, while column 51 contains only -8888's until such time as the operator overlays numbers (such as publication dates) of his choice.

Updating the Data Boxes

Updating of a data box--a process which involves either revising an already-existing row or creating, and entering data into, a new one--should be accomplished immediately upon publication of newly definitive data by SRS. Thus, updating typically involves one-or-more whole rows. In preparation for its receipt of the new data, a row to be updated should be set equal to -8888 at each location. Then, the year should be entered in column 50. (The first step can be accomplished readily in a loop for, say, row 43, by HENCEFORTH BOX GIVES BOXLIST(J);

BOX(43) = -8888. If the data box did not previously have a row 43, these commands will both create and initialize this row. In this connection, note that, each year, the operator should adjoin an additional variable (in this case, Y1981 = 43) to ROWYEARS.)

Actual data for entry into a data box should be entered into a one-dimensional array. A one-dimensional array of corresponding state column numbers, in the same order as the data to be entered, should also be prepared. (The latter task can be accomplished initially by running GIVENUMS and then entering the necessary state-name abbreviations, thereby defining an array of data-box column numbers corresponding to the state names entered. The array thus defined should be stored, since it will change rather little from year to year. In future years, after GIVENUMS has been run, the operator can easily delete state names from such an array by invoking RELCOMP. He can insert state names by adjoining them to the array and applying RANKER (assuming that, as is currently the case, SRS data is presented alphabetically by state, followed by U.S.).) By means of STATELST, subscripted by this state-name array, the operator can then read out the array in the form of (abbreviated) state names (as distinguished from column numbers) for purposes of verification. Thus, here again, he need not be concerned with column numbers.

Although, conceivably, as many as twenty-seven different state-name arrays (one for each data box) could be required for the updating of a given year's data, in practice a much smaller number is needed. For each crop under consideration, published SRS data for area harvested and yield typically refer to the same group of states, and for some crops, this group is used as well for the area-planted data.

If, say, CRNAPD81 ("corn, area planted, data, 1981") and CRNAPS81 ("corn, area planted, states, 1981") were the data- and state arrays defined as just

described, then the command CRNBXAP(Y1981,CRNAPS81) = CRNAPD81 would suffice to load the data. Of course, when many data boxes are being updated, loading is best done in a loop, using HENCEFORTH and OBJECT to facilitate looping through various array names.

Comments on the Design of the Economic Data Bank

The logical structure of the economic data bank--especially, the organizing of the data into a few, large, isomorphic two-dimensional arrays--confers many benefits. For example, corn production for all years and states and U.S. can be calculated with the single command CRNBXAH*CRNBXYD (though, preferably, the -8888's would first be switched to 0's by the use of WHERE). (Incidentally, SRS has, over time, used more than one rounding procedure in computing its production figures; thus, under any universal rounding scheme, the product of historical SRS area-harvested and yield figures need not equal ("digit-by-digit") the corresponding SRS production figure.) Another example of the usefulness of the data-bank design relates to the maintenance of data integrity. If, say, CRNBXAP has forty-three rows, then, by forming CRNBXAP*ARRAY(43 51: INTS(1 43*51)) and recording the SUM, SUMROWS, and SUMCOLS of this, one obtains an inexpensive but highly reliable means of detecting inadvertent changes in CRNBXAP. For, if the same computation were made at a later date, and, say, data had since been mistakenly transposed within CRNBXAP, the newly-computed values of the SUM, SUMROWS, and SUMCOLS would typically both indicate a change in CRNBXAP and help the operator locate the rows and columns (hence, the elements) which had changed. This procedure, done in a loop using BOXLIST, would help protect the integrity of the entire data bank. In a similar vein, one can check the accuracy of inputted area-planted and area-harvested figures by switching the -8888's in each AP and AH data box to 0's using WHERE, computing the SUMROWS

of the sub-array consisting of the first forty-eight columns, and comparing this (by means of the logical operator .NE.) to the forty-ninth (U.S.) column.

The Data-Manipulation Subsystem

Components of the Subsystem

The data-manipulation subsystem is a collection of Speakeasy objects and commands designed to work together to enable the System operator to access and manipulate the weather data, alone or in combination with the economic data, in an efficient and logically elegant manner. It consists of:

COEFFS - a keptlisted namelist of forty-eight column matrices, one for each state. The seven-character names consist of "COEFF" concatenated with a two-character state-name abbreviation (e.g., COEFFIL). Each of these matrices has as many elements as there are climatic divisions in the corresponding state, and (except for Nebraska) the ith matrix element is the ratio of the area of the ith climatic division of the state to the area of the state. (In the eight-element matrix COEFFNE, elements 1-3 correspond to climatic divisions 1-3, while elements 4-8 correspond to divisions 5-9.) The area ratios--known as "climatic-division area coefficients"--were provided by NOAA, rounded to four digits after the decimal point. The operator may wish to be aware that Rhode Island has only one climatic division, and that the rounded area coefficient of Florida's seventh climatic division ("Florida Keys") is 0.

Note that, for example, if PRCP1IL were a matrix whose columns, in order, were the time-series PRCP1IL01,..., PRCP1IL09, then the

command PRCP1IL*COEFFIL would produce a time-series of state average precipitation.

FIPSCODE - A 48 by 2 character array whose rows are the Federal Information Processing Standards (FIPS) codes of the forty-eight contiguous states. The rows correspond to the states ordered alphabetically, as in Figure 17. Each state FIPS code is a two-digit number used to denote that particular state. The array FIPSCODE is used only infrequently, usually to access data which is external to the Automated Weather/Yield System and whose naming-scheme uses FIPS codes.

GIVENUMS - a program which defines state-name (and U.S.) abbreviations as scalars. GIVENUMS defines AL = 1, AZ = 2, AR = 3,..., WY = 48, and US = 49 (cf. Figure 17).

PNAMLST - A namelist of forty-eight arrays--one for each of the contiguous states. Each array has twelve rows (corresponding to the twelve months; January corresponds to row 1, etc., in order) and as many columns as that state has climatic divisions (column 1 represents climatic division 1, etc., in order). Each array element is the thirty-year (1941-70) normal (i.e., average) precipitation for the corresponding climatic division, state, and month. The names of the arrays consist of the character string PNR (connoting "precipitation normal") followed by the appropriate two-character state-name abbreviation (e.g., PNRKS). See also TNAMLST.

PROB - A command created by entering HENCEFORTH PROB GIVES MEAN. The command PROB serves as the System's "probability operator." To understand its use, suppose TDATA is an n-element one-dimensional array of

temperatures (measured in degrees Fahrenheit), conceived of as being constructed from realizations of n independent and identically distributed copies of a temperature random variable, T. Suppose, further, that one wishes to find the probability that T exceeds 80. This can be estimated by N/n, where N is the number of elements of TDATA that exceed 80. However, N is merely SUM(TDATA.GT.80), so that N/n is merely MEAN(TDATA.GT.80), i.e., PROB(TDATA.GT.80).

STAREAS - a forty-eight-element one-dimensional array whose jth element is the area (in square miles) of the jth state (the states being ordered alphabetically, as in Figure 17).

STATELST - a 49 by 2 character array of state-name (and U.S.) abbreviations. The middle column of Figure 17 displays the first forty-eight rows of STATELST, while the forty-ninth row is the character array US.

TNAMLST - A namelist of forty-eight arrays. The description of TNAMLST is like that of PNAMLST, except that the arrays of TNAMLST contain temperature normals instead of precipitation normals, and the array names begin with T (for "temperature") instead of P.

UTREV - This program ("utility-revised") is the "master-control" program used in accessing and manipulating weather data. Examples of its use were given in Chapter 1. The operator should understand this program "inside and out." Toward this end, a line-by-line discussion of UTREV is now given. (Refer to Figure 19. The original line-numbering has been retained to facilitate comparison with other programs based on UTREV.)

```
EDITING UTREV
   1.0 PROGRAM
   1.2 GET PROMO; PROMO
   1.3 FREEIF ANSWERMT REVERSE
   1.5 GET STATELST
   1.6 DIGPRS=ARRAY(10 2:"01020304050607080910")
   1.7 COEFFS=KEPTLIST(COEFFS)
   2.0 AL=1;AZ=2;AR=3;CA=4;CO=5;CT=6;DE=7;FL=8;GA=9;ID=10;IL=11;IN=12
   3.0 IA=13;KS=14;KY=15;LA=16;ME=17;MD=18;MA=19;MI=20;MN=21;MS=22
   4.0 MO=23;MT=24;NE=25;NV=26;NH=27;NJ=28;NM=29;NY=30;NC=31;ND=32
   5.0 OH=33;OK=34;OR=35;PA=36;RI=37;SC=38;SD=39;TN=40;TX=41;UT=42
   6.0 VT=43;VA=44;WA=45;WV=46;WI=47;WY=48
   7.0 $$$$$$
   8.0 JAN=1;FEB=2;MAR=3;APR=4;MAY=5;JUN=6;JUL=7;AUG=8;SEP=9;OCT=10
   9.0 NOV=11;DEC=12
  10.0 ALL=-1000
  11.0 YES=1
  12.0 NO=0
  81.0 WETHRTYP="PDI"
  82.0 TEMP=1; PRCP=2; PDI=3
  83.0 WWW=ARRAY(2 4:"TEMPPRCP")
  90.0 "SPECIFY WEATHER TYPE (TEMP, PRCP, OR PDI)"
  91.0 REQUEST(WTYPE)
  92.0 IF (WTYPE.LT.3) WETHRTYP=WWW(WTYPE)
 110.0 "ENTER STATE NAME ABBREVIATIONS"
 120.0 REQUEST(STATNAMS)
 123.0 STATNAMS=ARRAY(STATNAMS)
 130.0 STATSCT=NOELS(STATNAMS)
 140.0 "ENTER ""ALL"" OR LIST CHOSEN CLIMATIC SUBDIVISIONS IN"
 150.0 "ORDER WITHOUT LEADING ZEROES"
 160.0 FOR STATNDX1=1 STATSCT
 170.0 REQUEST(OBJECT("SUBDVS" STATELST(STATNAMS(STATNDX1))))
 180.0 ENDLOOP STATNDX1
 190.0 "ENTER MONTHS IN ORDER"
 200.0 REQUEST(MONVALS)
 203.0 MONVALS=ARRAY(MONVALS)
 205.0 IF (SUM(MONVALS).LT.0) MONVALS=INTS(1 12)
 210.0 "ENTER STARTING AND ENDING YEARS FOR FIRST MONTH"
 220.0 REQUEST(STARTYR LASTYR)
 230.0 HOWMANY=NOELS(MONVALS)
 231.0 MONVAL1=MONVALS(1)
 231.5 MONVALSR=MONVALS
 232.0 WHERE (MONVALSR.LT.MONVAL1) MONVALSR=MONVALSR+12
 310.0 YEARSCT=LASTYR-STARTYR+1
 315.0 WHICHMON=INTS(1 HOWMANY)
 317.0 DDX=INTS(1 YEARSCT)
 318.0 MELD(WHICHMON DDX)
 322.0 TDLGF="WEAM"
 324.0 TDIP=STARTYR*1000+1
 326.0 TDLP=(LASTYR+(MONVALS(HOWMANY).LT.MONVAL1))*1000+MONVALS(HOWMANY)
```

```
 370 FOR STATNDX2=1 STATSCT
 380 STAT=STATELST(STATNAMS(STATNDX2))
 385 V111=WETHRTYP "1" STAT
 390 COEFFSTT=OBJECT("COEFF" STAT)
 400 SBDVNUM=NOELS(COEFFSTT)
 435 SUBDVS=OBJECT("SUBDVS" STAT)
 437 SUBDVS=ARRAY(SUBDVS)
 440 YESNEB=STATNAMS(STATNDX2).EQ.25
 445 YESALL=MIN(SUBDVS).EQ.-1000
 450 IF (YESALL.AND.(.NOT.YESNEB)) SUBDVS=INTS(1 SBDVNUM)
 460 IF (YESALL.AND.YESNEB) SUBDVS=1 2 3 5 6 7 8 9
 465 SBDVCT=NOELS(SUBDVS)
 470 SUBDVS2=SUBDVS
 480 IF (YESNEB) WHERE (SUBDVS2.GE.5) SUBDVS2=SUBDVS2-1
 490 STATCOL=0
 500 FOR JJ=1 SBDVCT
 510 V112=V111 DIGPRS(SUBDVS(JJ))
 520 STATCOL=TDAM(GET V112)*MEAN(COEFFSTT(SUBDVS2(JJ)))+STATCOL
 530 ENDLOOP JJ
 540 STATCOL=STATCOL/SUM(COEFFSTT(SUBDVS2))
 619 WHATIS STATCOL
 670 ANSWERMT(WHICHMON+HOWMANY*(DDX-1),STATNDX2)=
 680 &STATCOL(12*(DDX-1)+MONVALSR(WHICHMON))
 780 ENDLOOP STATNDX2
 785 FREE STATCOL
 790 WHATIS ANSWERMT
 795 REVERSE(MONVALS)=INTS(1 HOWMANY)
 800 "WANT TO COMBINE MONTHS?"
 810 CUMLABL:
 820 REQUEST(ANS)
 830 IF (ANS.EQ.0) GO TO TLABL
 840 "ENTER MONTHS TO BE COMBINED IN ORDER"
 850 REQUEST(CHOICES)
 853 CHOICES=ARRAY(CHOICES)
 860 ASKCHAR("WITHOUT BLANKS, ENTER NAME OF FORTHCOMING
 870 & ACCUMULATION ARRAY","PICKNAME=")
 880 HENCEFORTH CUMMAT GIVES OBJECT(PICKNAME)
 890 MONRANK=REVERSE(CHOICES)
 900 CHOICECT=NOELS(CHOICES)
 910 TRAJECT=HOWMANY*INTS(0 YEARSCT-1)
 920 CUMMAT=0
 930 FOR LL=1 CHOICECT
 940 CUMMAT=ANSWERMT(MONRANK(LL)+TRAJECT)+CUMMAT
 950 ENDLOOP LL
 955 IF (WTYPE.NE.2) CUMMAT=CUMMAT/CHOICECT
 960 WHATIS CUMMAT
 970 "WANT TO COMBINE MONTHS AGAIN?"
 980 GO TO CUMLABL
 990 TLABL:
*991 HENCEFORTH CUMMAT GIVES CUMMAT
 992 END
```

A Detailed Analysis of UTREV

Line 1.2: This line accesses and prints out the character array PROMO, stored in MYKEEP.

Line 1.3: The program defines the dimensions of ANSWERMT and REVERSE implicitly. If UTREV were run a second time with the old ANSWERMT and REVERSE still in the workspace, the new definitions could (if the new ANSWERMT or REVERSE were smaller) inadvertently leave part of the old object "sticking out" as part of the new one.

Line 1.5: STATELST will be used to build names with state-name abbreviations as a component, such as SUBDVSOH or COEFFIL.

Line 1.6: DIGPRS ("digit-pairs") will be used to build weather-variable names with climatic-division numbers as a component, such as PRCP1KS03.

Line 1.7: Selected coefficient matrices in COEFFS will be used to average weather records over a union of selected climatic divisions for each selected state.

Lines 2.0-6.0: Play the role of GIVENUMS.

Lines 8-9: Allow the operator to enter abbreviated month-names, while the computer registers the corresponding month-numbers, which are more useful for logical manipulation.

Lines 10-12: Allow the operator to enter ALL when prompted for a list of climatic divisions of a state, and to enter YES or NO later in the program, while the computer registers the corresponding numbers.

Lines 81-92: Prompt the operator to enter the type of weather data to be accessed and define the character array WETHRTYP ("weather type") as an aid in

the formation of weather-variable names. The operator responds with one of the character strings TEMP, PRCP, or PDI.

<u>Lines 110-120</u>: The operator responds to this prompt by typing his selected two-character state-name abbreviations, in any order. The states selected must be distinct. If more than the Speakeasy-allowable number of elements are to be entered into the array STATNAMS ("state names"), the operator should define an array, STATNAMS, of state numbers (with the use of GIVENUMS) prior to running UTREV and should reply to the STATNAMS prompt by entering STATNAMS.

<u>Line 123</u>: If only one state were entered into STATNAMS, STATNAMS would have been defined as a scalar, not an array. At the time the program was under development, and with the version of Speakeasy then available, this ambiguity in the structure of STATNAMS caused certain problems. (The author suspects--without recalling with certainty--that some needed commands (perhaps NOELS) could not then be applied to scalars.) Accordingly, this line was included in order to assure, unambiguously, an array structure for STATNAMS. Interestingly, in the Mu version of Speakeasy, a scalar structure for STATNAMS ceased to cause problems. However, in the just-released Pi version, scalars cannot be subscripted, so that line 123 is needed to assure an array structure for STATNAMS (cf. line 170).

<u>Line 130</u>: Counts the number of states selected. (STATSCT connotes "states-count").

<u>Lines 140-180</u>: Using a loop, these lines prompt the operator to select climatic divisions for each of his selected states. If, say, Illinois is one of the states selected, an array or scalar SUBDVSIL ("subdivisions-Illinois") of selected climatic-division numbers will be formed. The operator is permitted to enter ALL. In this case, SUBDVSIL (say) would be defined temporarily as

the scalar -1000--it would be redefined later (see lines 400-480). (STATNDX1 connotes "first state index.")

Lines 190-200: Prompt the operator to select months. He enters three-character abbreviations (JAN, OCT, etc., as given in lines 8-9) in the order desired, separated by blanks. (For example, if a September-May early-season precipitation index were to be formed, he would enter SEP OCT NOV DEC JAN FEB MAR APR MAY.) To select January through December, the operator may enter ALL. (MONVALS connotes "month values.")

Line 203: If only one month were selected, MONVALS would be a scalar, not an array. An array structure for MONVALS is needed (cf. line 231 and the discussion of line 123).

Line 205: Defines MONVALS in case the operator had entered ALL in line 200. (Recall that ALL = -1000. The "trick" here is to determine whether MONVALS contains -1000 (in which case it contains only -1000). Note that one could not in general inquire, say, whether MONVALS.EQ.-1000, since, if MONVALS had length exceeding 1, a structural error would result.)

Lines 210-220: Operator enters years desired.

Lines 230-232: (MONVALSR connotes "MONVALS-revised." HOWMANY is used later.) If the operator had entered, say, NOV DEC JAN MAR, MONVALS would equal (11 12 1 3). In order to aid in selecting the proper sequence of months from a monthly weather time-series, these lines would form the array (11 12 13 15). The need is to add 12 to each month-number in MONVALS which is exceeded in magnitude by any of the elements of MONVALS preceding it. The "trick" here is to note that an element of MONVALS is exceeded in this sense if and only if it is less than the first element of MONVALS.

Line 310: Counts the number of years selected (YEARSCT connotes "years-count").

Lines 315-318: (WHICHMON connotes "which month;" DDX connotes "index.") The key idea here may be illustrated as follows: Suppose one wished to employ a double loop over, say, indices i = 1, 2 and j = 1, 2, 3. Since subscripted arrays are more efficient than loops in Speakeasy, one could define arrays I = (1 2) and J = (1 2 3) and could then (using MELD) replicate them into redefined versions I = (1 1 1 2 2 2) and J = (1 2 3 1 2 3). Then, the six pairs I(k), J(k) (where k = 1,..., 6) would correspond to the six combinations of i, j which would appear in a double loop, and the redefined I and J could be used to accomplish in a single Speakeasy statement what would otherwise require a double loop. In the particular case at hand, the redefined WHICHMON and DDX will facilitate the selection of months and years in lines 670-680.

Lines 322-326: These lines select the TDAM logical group file WEA, the frequency M, the initial period (which is always taken to be January of STARTYR), and the last period. Note that the logical operator .LT. in line 326 is used to compare the first and last elements of (the original) MONVALS in order to determine whether the stream of months selected by the operator intersects two years. If it does, the logical object in line 326 takes the value 1 and insures the proper designation of the last TDAM period.

Lines 370-780: These lines constitute a major loop within UTREV. The loop (on STATNDX2, which connotes "second state index") defines the columns of a two-dimensional array ANSWERMT, one column for each of the states previously selected, ordered from left to right. (In an earlier version of this program, ANSWERMT--"answer matrix"--was defined as a matrix; hence its name. Although it is now an array, its name was retained for reasons of familiarity.) The rows of ANSWERMT correspond in increasing order to the month/year combinations

previously selected, arranged chronologically. The number of rows of ANSWERMT is HOWMANY*YEARSCT. Each element of ANSWERMT is an average of temperature, precipitation, or the Palmer Drought Index, over the region formed from the selected climatic divisions of the state in whose corresponding ANSWERMT column the element lies, for the month/year combination in whose corresponding ANSERMT row the element lies.

Lines 370-390: These lines start the loop over the selected states; define STAT as a state-name character array (e.g., if Illinois was the first state entered, then STAT would be the two-element character string "IL"); define the character array V111 as the first portion of a TDAM weather-variable name (e.g., as the seven-element character string "PRCP1IL" if PRCP and IL were entered); and define COEFFSTT ("coefficients for the state") as one of the column matrices in COEFFS (described earlier) (e.g., COEFFSTT would be COEFFIL if Illinois is the state being treated).

Lines 400-480: This section completes the task--begun in the loop of lines 140-180--of specifying which climatic divisions--and thus also which climatic-division area coefficients--are to be accessed for each selected state. There are two problems to be addressed: first, if, say, Illinois was selected and the operator entered ALL for his choice of climatic divisions, then SUBDVSIL was defined as -1000. In this case, it would be necessary now to redefine SUBDVSIL as INTS(1 SUBDVNUM), where SUBDVNUM ("subdivision number") is the total number of climatic divisions of Illinois. (Note that SUBDVNUM equals NOELS(COEFFIL).) The second problem to be addressed--not independently of the first--arises from the fact that Nebraska has no

climatic division 4, so that, if Nebraska were selected and all climatic divisions of it were requested _via_ ALL, then one could not merely define SUBDVSNE as INTS(1 SUBDVNUM), i.e., as INTS(1 8). Moreover, even if ALL were not entered for Nebraska but, rather, desired division-numbers for Nebraska were entered individually, including one or more of divisions 5, 6, 7, 8, or 9, then one would still need to use a revised version of SUBDVSNE in order to access the correct area coefficients from COEFFNE. For if, say, SUBDVSNE had originally been defined as (1 2 8 9), then the area coefficients corresponding to the TDAM variables (say) PRCP1NE01, PRCP1NE02, PRCP1NE08, PRCP1NE09 would be COEFFNE((1 2 7 8)), not COEFFNE((1 2 8 9)).

Line 400: Counts the total number of climatic divisions of the state being considered.

Line 435: Defines SUBDVS as the previously-defined subdivision list (e.g., as SUBDVSIL if IL is being considered).

Line 437: If only one climatic division were selected for the state corresponding to STAT, SUBDVS would be a scalar. Because a scalar cannot be subscripted in the Pi version of Speakeasy, an array structure for SUBDVS is needed (cf. line 510). Similarly, SUBDVS2 (cf. line 470) needs to inherit an array structure from SUBDVS (cf. line 520).

Lines 440 - 460: Ask (line 440) if Nebraska is the state currently under consideration (note that Nebraska is state number 25 in the standard System ordering) and (line 445) if ALL was entered for the current state's climatic divisions. In case ALL was entered, redefine (lines 450-460) the list SUBDVS of selected divisions. Next (line 465),

define SUBDVSCT ("subdivisions-count") as the number of divisions selected for the state. Finally (lines 470-480), make a copy SUBDVS2 of SUBDVS and, for the case of Nebraska, adjust downward (by 1) any elements of SUBDVS2 which are greater than or equal to 5, in order that the area coefficients in COEFFNE(SUBDVS2) correspond properly to the climatic divisions in SUBDVS.

Lines 490-540: This important section defines a single column STATCOL of ANSWERMT by first looping through the climatic divisions which were selected for the given state and then making a final adjustment in line 540. STATCOL is constructed recursively. It is initialized in line 490. Line 510 defines V112 (which, incidentally, like V111, has no particular connotation as a symbol) as a full TDAM variable name (e.g., TEMP1KS03) by adjoining the two-character division number to V111. Line 520, used recursively, constructs a weighted sum of the selected TDAM weather variables, the weights being the corresponding climatic-division area coefficients. (The role of MEAN here is to transform COEFFSTT(SUBDVS2(JJ)) from a one-element vector into a scalar, so that it can "legally" be multiplied by the array TDAM (GET V112).) (Note also that line 510 uses SUBDVS to access climatic divisions, while line 520 uses SUBDVS2 to access the corresponding area coefficients. At this point in the program, SUBDVS and SUBDVS2 differ if (and only if) some element of SUBDVS exceeds 3 and the state involved is Nebraska.) Finally, line 540 completes the construction of STATCOL by dividing by the sum of the area coefficients which were used previously as weights. To understand the need for this line (and indeed the entire rationale underlying the construction of STATCOL), suppose

that n climatic divisions with areas $A_1, \ldots, A_n$ have been selected from a state whose area is A, and let the corresponding climatic-division area coefficients be $C_1, \ldots, C_n$. Consider a single month, and suppose that the weather data corresponding to the n climatic divisions are $W_1, \ldots, W_n$. (W_1, say, could be the average temperature for the month over the first selected division). Then, as proved in [11], the weather datum (in the case suggested, the average temperature) over the entire region formed by the n selected climatic divisions is

$$
\begin{aligned}
&(W_1A_1 + \ldots + W_nA_n)/(A_1 + \ldots + A_n) \\
&= (W_1(A_1/A) + \ldots + W_n(A_n/A))/((A_1/A) + \ldots + (A_n/A)) \\
&= (W_1C_1 + \ldots + W_nC_n)/(C_1 + \ldots + C_n).
\end{aligned}
$$

It remains only to observe that, in effect, the numerator $(W_1C_1 + \ldots + W_nC_n)$ was constructed recursively by line 520, while line 540 merely divided by the denominator $(C_1 + \ldots + C_n)$ in order to form the desired ratio.

Line 619: Serves to reassure the operator that the program is running and is looping through the selected states. Each appearance of a response to this line's question indicates that a new column of ANSWERMT is about to be formed (see lines 670-680).

Lines 670-680: These lines define a column of ANSWERMT, using selected elements of STATCOL. Suppose, for example, that STARTYR is 1979, LASTYR is 1981, and the months of February and March were chosen at line 200. Then, STATCOL (which, it should be recalled, always begins with January of STARTYR) would be a twenty-seven-element array with elements corresponding, in order, to the twenty-seven months from

January, 1979 to March, 1981. ANSWERMT would have six rows, corresponding, in order, to February and March, 1979, February and March, 1980, and February and March, 1981. Note that these months correspond to coordinates 2, 3, 14, 15, 26, and 27 of STATCOL. Furthermore, here MONVALSR = (2, 3), DDX = (1 2 3 1 2 3), and WHICHMON = (1 1 1 2 2 2) (recall that DDX and WHICHMON were melded in line 318), so that 12*(DDX -1) + MONVALSR(WHICHMON) = (2 14 26 3 15 27). On the other hand, WHICHMON + HOWMANY*(DDX -1) = (1 3 5 2 4 6). Thus, row 1 of ANSWERMT corresponds to the second element of STATCOL, row 3 of ANSWERMT corresponds to the fourteenth element of STATCOL, etc., as is desired. (Note that (1 3 5 2 4 6) and (2 14 26 3 15 27) are merely permutations--by means of the <u>same</u> permutation operator--of (1 2 3 4 5 6) and (2 3 14 15 26 27), respectively.)

<u>Line 780</u>: Ends the loop on STATNDX2.

<u>Line 785</u>: Guards against the possibility that the right-hand side of the equation in line 520 could use an "old," unintended value of STATCOL in the event that, for some reason (e.g., a computer malfunction), the loop in lines 500-530 were run without the benefit of line 490.

<u>Line 790</u>: Signals to the operator that ANSWERMT has been defined.

<u>Lines 795-990</u>: Permit the operator to aggregate the data in ANSWERMT over months (correspondingly, over rows of ANSWERMT) of his choice, enabling him to form new data such as January-February average temperature or total precipitation. The process has the structure of an implicit loop; the operator is repeatedly queried (lines 800-850) for choices of months to be combined, and for names to be assigned to the arrays of combined data which will be

created (lines 860-870). The operator exits from the loop by replying NO to the prompt in line 820.

As an example, suppose that MONVALS = (12 1 2 3), STARTYR = 1979, LASTYR = 1981, and that the operator wishes to calculate total December-January precipitation. Note that, here, ANSWERMT would have twelve rows (corresponding to December, January, February, and March, for three years), and the desired December-January totals would be obtained by adding rows 1 and 2, 5 and 6, and 9 and 10. In the following explanations of individual lines, this example will continue to be used as a point of reference.

Line 795: This line begins to attack the logical problem of determining which rows of ANSWERMT are to be combined in response to the choice of months to be made in line 850. REVERSE is implicitly defined as a one-dimensional array whose length is the largest integer in MONVALS, and whose values at the coordinates represented by the elements of MONVALS are set equal to the elements of INTS(1 HOWMANY). In the present example, REVERSE, subscripted by (12 1 2 3), is set equal to (1 2 3 4), so that the result is REVERSE = (2 3 4 0 0 0 0 0 0 0 0 1). The "trick" here is that REVERSE tells us, for each originally selected month, whether that month was the first, or second, or third, etc., element of MONVALS; that is, REVERSE implicitly "ranks" the elements of MONVALS. Thus, for example, the fact that "March" (recall MAR = 3) is the fourth element of MONVALS is represented by the fact that REVERSE(MAR) = 4. Similarly, REVERSE(DEC) = 1, REVERSE(JAN) = 2, and REVERSE(FEB) = 3.

Lines 800-830: The operator enters YES or NO (cf. lines 11-12). If NO is entered, line 830 transfers control to line 990, outside the loop.

If YES is entered, line 810 serves later as a return point for the loop. Note that the word "combine" in line 800 is neutral, referring either to totalling (as for precipitation) or averaging (as for temperature). (In line 810, CUMLABL connotes "accumulation label." TLABL in line 830 just means a "label," while ANS in line 830 of course connotes "answer.")

Lines 840-853: Prompt the operator to choose months to be combined. The months chosen must be among those selected in line 200 and must be entered here in the same relative order. A single month may be chosen, if desired. In the present example, the operator would enter DEC JAN.

For an explanation of the imposition of an array structure on CHOICES, refer to the discussion of line 123 (and note that NOELS is used in line 900).

Lines 860-870: Prompt the operator to enter a Speakeasy name of eight-or-fewer characters.

Line 880: Permits references to CUMMAT to invoke the name which the operator just entered.

Line 890: MONRANK ranks the elements of CHOICES. In the present example, MONRANK = (1 2), corresponding to the fact that December and January--the months chosen to be combined--were first and second in the list of months entered at line 200.

Line 900: CHOICECT ("choice-count") counts the number of months to be combined.

Line 910: TRAJECT ("trajectory") is used in line 940 as an aid in selecting the rows of ANSWERMT which are to be added. See the discussion for line 940.

<u>Line 920</u>: Initializes CUMMAT ("accumulation matrix") in preparation for the loop of lines 930-950. (In an earlier version of UTREV, CUMMAT had a matrix structure. Although it is no longer a matrix in the present version, its name has been retained for reasons of familiarity.)

<u>Lines 930-950</u>: These lines constitute a loop which, in effect, is over the months chosen for aggregation. For each such month and corresponding index value LL, the array ANSWERMT(MONRANK(LL) + TRAJECT) consists of those rows of ANSWERMT which represent that month. The loop forms one such array for each such month and recursively constructs their sum. Consider again our example for the case LL = 1; since MONRANK(1) = 1 and TRAJECT = 4*INTS(0 2) = (0 4 8), we have MONRANK(1) + TRAJECT = (1 5 9). Similarly, MONRANK(2) + TRAJECT = (2 6 10). Thus, to calculate the total December-January precipitation for 1979-81, the loop adds ANSWERMT(1 5 9) to ANSWERMT(2 6 10), i.e., it adds rows 1 and 2, 5 and 6, and 9 and 10, of ANSWERMT.

<u>Line 955</u>: Although the goal in aggregating precipitation figures over various months is to calculate the <u>total</u> precipitation, the goal in aggregating temperature- or Palmer Drought Index figures is to calculate their <u>average</u>. This line completes the averaging process for the latter two cases.

<u>Line 960</u>: Signals the operator that CUMMAT has been defined. CUMMAT has one row for each year from STARTYR to LASTYR, and each row contains an aggregation of data for the months chosen in line 850. The columns of CUMMAT, of course, represent the same states and

climatic divisions as the corresponding columns of ANSWERMT. Note that CUMMAT is still controlled here by the HENCEFORTH of line 880.

Lines 970-980: Ask the operator if he wishes to conduct another aggregation, using (presumably) a different choice of months. Control is returned to line 810. If the operator enters YES at line 820, he continues through the loop; if he answers NO, control passes to line 990, and the program soon ends.

Line 990: Serves as a Speakeasy label.

Line 991: It is a prudent policy to neutralize HENCEFORTH statements when they are no longer needed. This line neutralizes the HENCEFORTH which would remain from the last execution of line 880, if indeed line 880 was executed.

Line 992: Ends the program UTREV. Because of differences in significant digits between weekly and monthly NOAA data and for other technical reasons, the data-arrays generated by UTREV are not rounded off; this step is left to the operator to accomplish according to individual circumstances.

Weighting Weather Data

Weather series generated by UTREV may readily be weighted (by area planted, population, herd size, etc.). To accomplish this, the operator should insert a new line in UTREV, immediately following line 790, containing a PAUSE command. When the PAUSE is executed, the command ANSWERMT = SUMROWS(ANSWERMT*WEIGHTS) (where WEIGHTS is a previously-defined array of the same dimensions as ANSWERMT) redefines ANSWERMT as a weighted version of its original self. Here, WEIGHTS is assumed to consist of nonnegative elements, and each row is assumed to sum to 1. The operator may now continue the execution of the

program and is free to use the aggregation capabilities made available in the latter part of UTREV.

The Automated Forecasting Subsystem

The automated yield-forecastinng subsystem is a collection of four Speakeasy programs--CRNPROGN, SOYPROGN, SRGPROGN, and BARPROGN--which generate forecasts of U.S. crop yields for corn, soybeans, sorghum, and barley, respectively. The programs are based on linear regression models reflecting both economic factors, such as fertilizer prices, and "weather-up-to-the-present." They may be run in the interactive mode or in batch. The user merely enters the Speakeasy workspace with, say, a size of 175, gets the desired program from MYKEEP storage, and executes the program by invoking its name. The program generates a printout displaying the forecast and supporting information, as described on pp.24-9.

It must be recognized that economic forecasting is an activity which calls for a certain attitude of humility on the part of its practitioners, and an econometric model which is used today may be updated, modified, or even replaced tomorrow. Thus, in the author's view, although the particular econometric models used in the automated forecasting subsystem may not be without interest in their own right, the importance of the subsystem lies not so much in the models themselves as in the development of a methodology which permits the inclusion of objective information on "weather-up-to-the-present" in an automated system for crop-yield forecasting. This methodology is very robust with respect to changes in the models used.

Since the four forecasting programs are all conceptually similar, any one of them may be analyzed in order to understand the underlying methodology. Toward this end, a discussion of CRNPROGN is now given. (Refer to Figures 20,

Figure 20--Listing of program UTREVBEG

```
EDITING UTREVBEG
  1.0 PROGRAM
  1.3 FREEIF ANSWERMT REVERSE
  1.5 GET STATELST
  1.6 DIGPRS=ARRAY(10 2:"01020304050607080910")
  1.7 COEFFS=KEPTLIST(COEFFS)
  2.0 AL=1;AZ=2;AR=3;CA=4;CO=5;CT=6;DE=7;FL=8;GA=9;ID=10;IL=11;IN=12
  3.0 IA=13;KS=14;KY=15;LA=16;ME=17;MD=18;MA=19;MI=20;MN=21;MS=22
  4.0 MO=23;MT=24;NE=25;NV=26;NH=27;NJ=28;NM=29;NY=30;NC=31;ND=32
  5.0 OH=33;OK=34;OR=35;PA=36;RI=37;SC=38;SD=39;TN=40;TX=41;UT=42
  6.0 VT=43;VA=44;WA=45;WV=46;WI=47;WY=48
  7.0 $$$$$$
  8.0 JAN=1;FEB=2;MAR=3;APR=4;MAY=5;JUN=6;JUL=7;AUG=8;SEP=9;OCT=10
  9.0 NOV=11;DEC=12
 10.0 ALL=-1000
 11.0 YES=1
 12.0 NO=0
 81.0 WETHRTYP="PDI"
 82.0 TEMP=1; PRCP=2; PDI=3
*83.0 WWW=ARRAY(2 4:"TEMPPRCP")
 84.0 END
```

21, and 22.) Since CRNPROGN is based on concepts used in UTREV, the discussion will assume a knowledge of UTREV. (At this point, the reader may wish to review Figure 19 and the analysis of UTREV given on pp.47-59.)

A Detailed Analysis of CRNPROGN

Line 2: The keptlist CRNSTUFF contains the six-column regressor matrix CCC and the six-element array COEFF of corresponding regression coefficients. Optional material related to the econometric model, such as fitted values, may also be stored in CRNSTUFF. In this particular linear regression model, U.S. corn yield is regressed on a constant term, a linear time (really, technology) trend, the U.S. area planted for corn, the July precipitation over the region composed of Illinois, Indiana, Iowa, Ohio, and Nebraska, and dummy variables for 1970 and 1972.

Lines 3-4: The program UTREVBEG ("UTREV beginning;" see Figure 20) is simply the beginning part of UTREV and serves the same purpose. While not all of UTREVBEG is technically required in CRNPROGN, its use in all of the forecasting programs contributes to flexibility and conceptual simplicity.

Lines 5-20: The program UTREVEND ("UTREV end;" see Figure 21) which is referred to in lines 19-20 essentially accomplishes what the remainder of UTREV accomplishes after the user has replied to the various prompts called for in UTREV. However, UTREVEND is designed to accept replies that are written into a program rather than given interactively by the user. These replies, which, in effect, answer questions to be asked implicitly in line 20, appear in lines 5-18. Lines 5-15 are straightforward. Line 16 (where "CHOI" connotes "choice") corresponds to lines 840-850 of UTREV, while line 17 corresponds to lines 860-870 of UTREV. Line 18 (where COMBINCT connotes "combination count") specifies how

Figure 21--Listing of program UTREVEND

```
EDITING UTREVEND
   1.0 PROGRAM
  90.0 "SPECIFY WEATHER TYPE (TEMP, PRCP, OR PDI)"
  92.0 IF (WTYPE.LT.3) WETHRTYP=WWW(WTYPE)
 110.0 "ENTER STATE NAME ABBREVIATIONS"
 123.0 STATNAMS=ARRAY(STATNAMS)
 130.0 STATSCT=NOELS(STATNAMS)
 140.0 "ENTER ""ALL"" OR LIST CHOSEN CLIMATIC SUBDIVISIONS IN"
 150.0 "ORDER WITHOUT LEADING ZEROES"
 190.0 "ENTER MONTHS IN ORDER"
 203.0 MONVALS=ARRAY(MONVALS)
 205.0 IF (SUM(MONVALS).LT.0) MONVALS=INTS(1 12)
 210.0 "ENTER STARTING AND ENDING YEARS FOR FIRST MONTH"
 230.0 HOWMANY=NOELS(MONVALS)
 231.0 MONVAL1=MONVALS(1)
 231.5 MONVALSR=MONVALS
 232.0 WHERE (MONVALSR.LT.MONVAL1) MONVALSR=MONVALSR+12
 310.0 YEARSCT=LASTYR-STARTYR+1
 315.0 WHICHMON=INTS(1 HOWMANY)
 317.0 DDX=INTS(1 YEARSCT)
 318.0 MELD(WHICHMON DDX)
 322.0 TDLGF="WEAM"
 324.0 TDIP=STARTYR*1000+1
 326.0 TDLP=(LASTYR+(MONVALS(HOWMANY).LT.MONVAL1))*1000+MONVALS(HOWMANY)
 370.0 FOR STATNDX2=1 STATSCT
 380.0 STAT=STATELST(STATNAMS(STATNDX2))
 385.0 V111=WETHRTYP "1" STAT
 390.0 COEFFSTT=OBJECT("COEFF" STAT)
 400.0 SBDVNUM=NOELS(COEFFSTT)
 435.0 SUBDVS=OBJECT("SUBDVS" STAT)
 437.0 SUBDVS=ARRAY(SUBDVS)
 440.0 YESNEB=STATNAMS(STATNDX2).EQ.25
 445.0 YESALL=MIN(SUBDVS).EQ.-1000
 450.0 IF (YESALL.AND.(.NOT.YESNEB)) SUBDVS=INTS(1 SBDVNUM)
 460.0 IF (YESALL.AND.YESNEB) SUBDVS=1 2 3 5 6 7 8 9
 465.0 SBDVCT=NOELS(SUBDVS)
 470.0 SUBDVS2=SUBDVS
```

(UTREVEND concluded)

```
 480 IF (YESNEB) WHERE (SUBDVS2.GE.5) SUBDVS2=SUBDVS2-1
 490 STATCOL=0
 500 FOR JJ=1 SBDVCT
 510 V112=V111 DIGPRS(SUBDVS(JJ))
 520 STATCOL=TDAM(GET V112)*MEAN(COEFFSTT(SUBDVS2(JJ)))+STATCOL
 530 ENDLOOP JJ
 540 STATCOL=STATCOL/SUM(COEFFSTT(SUBDVS2))
 619 WHATIS STATCOL
 670 ANSWERMT(WHICHMON+HOWMANY*(DDX-1),STATNDX2)=
 680 &STATCOL(12*(DDX-1)+MONVALSR(WHICHMON))
 780 ENDLOOP STATNDX2
 785 FREE STATCOL
 790 WHATIS ANSWERMT
 795 REVERSE(MONVALS)=INTS(1 HOWMANY)
 800 "WANT TO COMBINE MONTHS?"
 830 IF (ANS.EQ.0) GO TO TLABL
 835 FOR CC=1 COMBINCT
 840 "ENTER MONTHS TO BE COMBINED IN ORDER"
 853 CHOICES=ARRAY(OBJECT("CHOI" CC))
 860 "WITHOUT BLANKS, ENTER NAME OF FORTHCOMING ACCUMULATION ARRAY"
 875 PICKNAME=OBJECT("PICK" CC)
 880 HENCEFORTH CUMMAT GIVES OBJECT(PICKNAME)
 890 MONRANK=REVERSE(CHOICES)
 900 CHOICECT=NOELS(CHOICES)
 910 TRAJECT=HOWMANY*INTS(0 YEARSCT-1)
 920 CUMMAT=0
 930 FOR LL=1 CHOICECT
 940 CUMMAT=ANSWERMT(MONRANK(LL)+TRAJECT)+CUMMAT
 950 ENDLOOP LL
 955 IF (WTYPE.NE.2) CUMMAT=CUMMAT/CHOICECT
 960 WHATIS CUMMAT
 980 ENDLOOP CC
 990 TLABL:
*991 HENCEFORTH CUMMAT GIVES CUMMAT
 992 END
```

Figure 22--Listing of program CRNPROGN

```
EDITING CRNPROGN
PROGRAM
CRNSTUFF=KEPTLIST(CRNSTUFF)
GET UTREVBEG
EXECUTE UTREVBEG
WTYPE=PRCP
STATNAMS=IL IN IA OH NE
SUBDVSIL=ALL
SUBDVSIN=ALL
SUBDVSIA=ALL
SUBDVSOH=ALL
SUBDVSNE=ALL
MONVALS=JUL
STARTYR=1951
LASTYR=1980
ANS=YES
CHOI1=JUL
PICK1="CR1"
COMBINCT=1
GET UTREVEND
EXECUTE UTREVEND
AREAS=KEPTLIST(AREAS)
CRPJ=MFAM(CR1)
ARCP=MATRIX(5,1:AREAIL,AREAIN,AREAIA,AREAOH,AREANE)
CRPJUL=(CRPJ*ARCP)/SUM(ARCP)
CCC(,4)=AFAM(CRPJUL)
CRNCAST=CCC*VFAM(COEFF)
CRNCAST=INTPART(AFAM(CRNCAST)*10+.5)/10
TOP="U.S. CORN YIELD FORECAST"
DAT=DATE
WHATIS CRNCAST
$$
$$
TDLGF="WEAA"
TDIP=1901001; TDLP=1952001
```

(CRNPROGN concluded)

```
 30.5 LSTMXDWK=MAX(TDAM(GET "WKMXFLG"))
 30.6 GET ENDWK80
 30.7 WETHBDRY=ENDWK80(LSTMXDWK)
 30.8 TYPE("NOTE:  FOR THIS FORECAST, WEATHER WAS AVAILABLE THROUGH
 30.9 & NOAA WEEK" LSTMXDWK)
 31.0 $$
 31.1 $$
 31.2 SPACE 2
 32.0 "*****************************************************************************"
 33.0 TABULATE (INTS(1951 1980) CRNCAST:TITLE TOP)
 34.0 SPACE 3
 35.0 "MAJOR ECONOMIC ASSUMPTIONS:"
 36.0 "U.S. PLANTED ACREAGE OF CORN FOR 1980: 83.478 MIL."
 37.0 SPACE 1
 38.0 "THIS FORECAST MADE ON" DAT "WITH WEATHER AVAILABLE THROUGH" WETHBDRY
 39.0 "*****************************************************************************"
 40.0 SPACE 2
 41.0 $$
 42.0 $$
 43.0 GET CRNTK
 44.0 NOROWSTK=NOROWS(CRNTK)
 45.0 CRNTK(NOROWSTK+1)=LSTMXDWK CRNCAST(NOELS(CRNCAST))
 46.0 & AFAM(CCC(NOROWS(CCC)))*AFAM(COEFF)
 48.0 IMPACTS=NAMELIST(NOAAWEEK CRNCAST CONST TREND CRNAP JULPRCP DUM70 DUM72)
 48.1 LASTCAST=CRNTK(NOROWSTK)
 48.2 THISCAST=CRNTK(NOROWSTK+1)
 48.3 DIF2MIN1=THISCAST-LASTCAST
 49.0 TABULATE(IMPACTS LASTCAST THISCAST DIF2MIN1)
 50.0 SPACE 2
 50.5 VARVALS=CCC(NOROWS(CCC))
 51.0 VARVALS COEFF
*52.0 KEEP CRNTK
 53.0 END
```

many accumulation arrays are to be formed, i.e., how many times the implicit loop of lines 800-990 of UTREV is to be traversed.

Line 21: AREAS is a keptlist of state areas defined as scalars. For example, the scalar AREAIL in AREAS is the area of Illinois (in square miles). (AREAS fulfilled the function of STAREAS (p.44) before the latter object was introduced into the System.) The state areas will be used to weight state weather data so as to form a regional average.

Lines 22-25: These lines compute a time-series of average July precipitation over the region formed by Illinois, Indiana, Iowa, Ohio, and Nebraska and define the fourth column of the regressor matrix CCC as that time-series. Note that CR1 is the five-column array of state precipitation data for July which was created by UTREVEND. The weighting of the state precipitation values by the state areas is accomplished in line 24 with the use of matrix multiplication. (The array CR1 was changed into a matrix, CRPJ, in line 22 by the command MFAM.)

Lines 26-27: These lines multiply the regressor matrix (which now contains the newly-computed weather data) by the vector of regression coefficients to arrive at an annual time-series of crop-yield estimates of which the last is the current forecast. Line 27 rounds off the estimates. An entire time-series (rather than simply the current forecast) is computed as a check against computer malfunctions and similar misfortunes.

Lines 28-42: These lines set the stage for, and execute, the first part of the forecast printout. Line 28 defines a TITLE for use in the TABULATE statement of line 33. Line 29 permits the current day's date to be printed out at line 38. Line 30.5 defines LSTMXDWK ("last mixed week") as the largest

element of the TDAM variable WKMXFLG (see p.35); thus, LSTMXDWK is the NOAA week number of the last week of weather data which has been used to update the weather data bank. This is printed out at lines 30.8-30.9. Line 30.6 gets ENDWK80 ("ends of weeks for 1980"), which is a character array whose rows display the calendar dates of the last day of each of the NOAA-numbered weeks for the current crop year (which, in the case at hand, was 1980.) In line 30.7, ENDWK80 is subscripted by the week number (equivalently: row number) LSTMXDWK, so that WETHBDRY ("weather boundary") is a character array displaying the calendar date up to which the weather data bank has been updated. It is printed out at line 38.

Lines 43-49: These lines accomplish the printing out of detailed information regarding the current forecast and the previous forecast (see pp.26-9). CRNTK ("corn track") is an eight-column two-dimensional array that is stored in MYKEEP and serves as an historical record of the weekly forecasts and related information. It is initialized and stored at the start of the forecasting season as an array of zeroes and gains a new (non-zero) row each time a forecast is made (typically, weekly, as new weeks of weather data are received). Line 43 gets CRNTK from MYKEEP and line 44 defines NOROWSTK ("number of rows of corn track") as what its name implies. Line 45 adjoins a new row to CRNTK. The eight elements of this row are, respectively, the NOAA week number of the latest week of weather data which has been mixed into the weather data bank; the current crop-yield forecast (which, it should be recalled, is the last element of CRNCAST); and six "impacts" which are obtained by multiplying (as arrays) the last row of the six-column regressor matrix CCC by the six-element array COEFF of regression coefficients. For use in the TABULATE command of line 49, line 48 defines a namelist, IMPACTS,

whose last six elements refer to the regressors, namely: the constant term; the time-trend (technology-trend) variable; corn acreage planted; July precipitation; and dummy variables for 1970 and 1972. Lines 48.1-48.2 define LASTCAST ("last-forecast information") and THISCAST ("this-forecast information") as the next-to-last row, and last row, respectively, of the newly-expanded CRNTK, while line 48.3 defines DIF2MIN1 ("difference; item 2 minus item 1") as the difference of these rows. Thus, the first element of DIF2MIN1 is the number of weeks of additional weather data being used since the previous forecast; the second element is the difference in the forecasts proper; and elements 3-8 are the difference in the regressor impacts between the current and previous forecasts. Finally, line 49 prints out information regarding the current and previous forecasts. Note, in particular, that THISCAST breaks down the current forecast into components ("impacts") attributable to each of the regressors, while DIF2MIN1 breaks down the change between the current and previous forecasts into changes in these impacts.

Lines 50-51: These lines print out the current values of the regressor variables and the corresponding regression coefficients, ordered the same as the last six elements of IMPACTS (see line 48).

Line 52: Keeps CRNTK (newly-enlarged by one row) in MYKEEP in readiness for the next forecast.

Line 53: Ends the program.

The programs SOYPROGN, SRGPROGN, and BARPROGN are conceptually similar to CRNPROGN and are listed in the Appendix.

APPENDIX

Figures 23-25 are listings of the Speakeasy programs SOYPROGN, SRGPROGN, and BARPROGN, which constitute part of the automated forecasting subsystem. Since these programs are conceptually similar to CRNPROGN (discussed on pp.62, 67-9), a detailed analysis of them is omitted. Figure 26 lists a program, MCQPROGN, which generates state corn-yield forecasts (based on "weather-up-to-the-present") for Illinois, Indiana, Iowa, Missouri, and Ohio. This program is based on regression models developed by James McQuigg (cf. [3], pp.60-2, for an example of the modelling approach used by McQuigg.) (Note: The object MCQARRAY referred to in MCQPROGN is a 10 by 35 array containing regression coefficients (cf. [3], p.61).) Figure 27 displays the type of forecast printout which is generated by MCQPROGN. Finally, Figure 28 lists the program MIXPROG, which was discussed on pp.34-5.

Figure 23--Listing of program SOYPROGN

```
EDITING SOYPROGN
PROGRAM
SOYSTUFF=KEPTLIST(SOYSTUFF)
GET UTREVBEG
EXECUTE UTREVBEG
WTYPE=PRCP
STATNAMS=OH IN IL IA NE
SUBDVSOH=ALL
SUBDVSIN=ALL
SUBDVSIL=ALL
SUBDVSIA=ALL
SUBDVSNE=ALL
MONVALS=SEP OCT NOV DEC JAN FEB MAR APR MAY JUL AUG
STARTYR=1950
LASTYR=1979
ANS=YES
CHOI1=SEP OCT NOV DEC JAN FEB MAR APR MAY
PICK1="SEPMAY"
CHOI2=JUL AUG
PICK2="JULAUG"
COMBINCT=2
GET UTREVEND
EXECUTE UTREVEND
DENOM1=778305
DENOM2=239629
MT=MAT(5 1:3236.7 5104.7 10464 10931 5077.7)
SMINDEX=MFAM(SEPMAY)*MT/DENOM1
SMINDEX=INTPART(AFAM(SMINDEX)*10**4+.5)/10**4
WHATIS SMINDEX
JAINDEX=MFAM(JULAUG)*MT/DENOM2
JAINDEX=INTPART(AFAM(JAINDEX)*10**4+.5)/10**4
WHATIS JAINDEX
VVV(NOROWS(VVV),4)=SMINDEX(NOELS(SMINDEX))
VVV(NOROWS(VVV),5)=JAINDEX(NOELS(JAINDEX))
SOYCAST=VVV*COEFF
SOYCAST=INTPART(AFAM(SOYCAST)*10+.5)/10
TOP="U.S. SOYBEAN YIELD FORECAST"
DAT=DATE
WHATIS SOYCAST
$$
$$
TDLGF="WEAA"
TDIP=1901001; TDLP=1952001
LSTMXDWK=MAX(TDAM(GET "WKMXFLG"))
GET ENDWK80
WETHBDRY=ENDWK80(LSTMXDWK)
TYPE("NOTE:  THIS FORECAST USES WEATHER THROUGH NOAA WEEK" LSTMXDWK)
$$
$$
SPACE 2
```

(SOYPROGN concluded)

```
"*****************************************************************************"
TABULATE(INTS(1951 1980) SOYCAST: TITLE TOP)
SPACE 3
"ECONOMIC ASSUMPTIONS:"
"U.S. HARVESTED ACREAGE OF SOYBEANS FOR 1980:  69.187 MIL."
"ESTIMATE OF ESCS/STATISTICS FERTILIZER INDEX FOR 1980: 248"
SPACE 1
"THIS FORECAST MADE ON" DAT "USING WEATHER THROUGH" WETHBDRY
"*****************************************************************************"
SPACE 2
$$
$$
GET SOYTK
NOROWSTK=NOROWS(SOYTK)
SOYTK(NOROWSTK+1)=LSTMXDWK SOYCAST(NOELS(SOYCAST))
& AFAM(VVV(NOROWS(VVV)))*AFAM(COEFF)
IMPACTS=NAMELIST(NOAAWEEK SOYCAST CONST SOYSH FERTM INDEXSM
& INDEXJA DUM74 TREND)
LASTCAST=SOYTK(NOROWSTK)
THISCAST=SOYTK(NOROWSTK+1)
DIF2MIN1=THISCAST-LASTCAST
TABULATE(IMPACTS LASTCAST THISCAST DIF2MIN1)
SPACE 2
VARVALS=VVV(NOROWS(VVV))
VARVALS COEFF
*KEEP SOYTK
END
```

Figure 24--Listing of program SRGPROGN

```
EDITING SRGPROGN
PROGRAM
SRGSTUFF=KEPTLIST(SRGSTUFF)
GET UTREVBEG
EXECUTE UTREVBEG
WTYPE=PRCP
STATNAMS=TX KS NE MO OK
SUBDVSTX=1 2 3 7
SUBDVSKS=ALL
SUBDVSNE=3 5 6 7 8 9
SUBDVSMO=3 4
SUBDVSOK=1 2 3 4 5 6 7
MONVALS=SEP OCT NOV DEC JAN FEB MAR APR MAY JUN JUL AUG
STARTYR=1945
LASTYR=1979
ANS=YES
CHOI1=SEP OCT NOV DEC JAN FEB MAR APR MAY
PICK1="PSEPMAY"
CHOI2=JUN JUL AUG
PICK2="PJUNAUG"
COMBINCT=2
GET UTREVEND
EXECUTE UTREVEND
GET SRGBXAP
AAA=SRGBXAP(INTS(8 41),(TX KS NE MO OK))
AAA(35)=SRGBXAP(42,(TX KS NE MO OK))
DDSEPMAY=SUMROWS(PSEPMAY*AAA)/SUMROWS(AAA)
DDJUNAUG=SUMROWS(PJUNAUG*AAA)/SUMROWS(AAA)
FREEIF ANSWERMT REVERSE
WTYPE=TEMP
```

(SRGPROGN concluded)

```
25.0 MONVALS=JAN JUN
25.5 STARTYR=1946
25.7 LASTYR=1980
26.0 CHOI1=JAN
27.0 PICK1="TJAN"
28.0 CHOI2=JUN
29.0 PICK2="TJUN"
30.0 EXECUTE UTREVEND
31.0 DDJANT=SUMROWS(TJAN*AAA)/SUMROWS(AAA)
32.0 DDJUNT=SUMROWS(TJUN*AAA)/SUMROWS(AAA)
34.0 V(35,INTS(2 8))=1 2.33 DDSEPMAY(35) DDJUNAUG(35) DDJANT(35) DDJUNT(35) 52.95
35.0 SRGCAST=V(,INTS(2 8))*VFAM(COEFF)
36.0 SRGCAST=INTPART(AFAM(SRGCAST)*10+.5)/10
37.0 TOP="U.S. SORGHUM YIELD FORECAST"
38.0 DAT=DATE
38.5 WHATIS SRGCAST
38.6 $$
38.7 $$
39.1 TDLGF="WEAA"
39.2 TDIP=1901001; TDLP=1952001
39.3 LSTMXDWK=MAX(TDAM(GET "WKMXFLG"))
39.5 GET ENDWK80
39.6 WETHBDRY=ENDWK80(LSTMXDWK)
39.7 TYPE("NOTE:  THIS FORECAST USES WEATHER THROUGH NOAA WEEK" LSTMXDWK)
39.8 $$
39.9 $$
40.0 SPACE 2
41.0 "****************************************************************************"
42.0 TABULATE(INTS(1946 1980) SRGCAST :TITLE TOP)
43.0 SPACE 3
44.0 "MAJOR ECONOMIC ASSUMPTIONS:"
46.0 "SORGHUM PRICE TO FARMERS IN 1979:  $2.33"
47.0 SPACE 1
48.0 "THIS FORECAST MADE ON" DAT "USING WEATHER THROUGH" WETHBDRY
49.0 "****************************************************************************"
50.0 SPACE 2
51.0 $$
52.0 $$
53.0 GET SRGTK
54.0 NOROWSTK=NOROWS(SRGTK)
55.0 SRGTK(NOROWSTK+1)=LSTMXDWK SRGCAST(NOELS(SRGCAST))
56.0 & AFAM(V(NOROWS(V),INTS(2 8)))*AFAM(COEFF)
58.0 IMPACTS=NAMELIST(NOAAWEEK SRGCAST CONST SRGRPF1 DDSEPMAY DDJUNAUG
59.0 & DDJANT DDJUNT TREND)
59.1 LASTCAST=SRGTK(NOROWSTK)
59.2 THISCAST=SRGTK(NOROWSTK+1)
59.3 DIF2MIN1=THISCAST-LASTCAST
60.0 TABULATE(IMPACTS LASTCAST THISCAST DIF2MIN1)
61.0 SPACE 2
61.5 VARVALS=V(NOROWS(V),INTS(2 8))
62.0 VARVALS COEFF
*63.0 KEEP SRGTK
64.0 END
```

Figure 25--Listing of program BARPROGN

```
EDITING BARPROGN
  1.0 PROGRAM
  2.0 BARSTUFF=KEPTLIST(BARSTUFF)
  3.0 GET UTREVBEG
  4.0 EXECUTE UTREVBEG
  5.0 WTYPE=TEMP
  6.0 STATNAMS=MN MT ND
  7.0 SUBDVSMN=1
  8.0 SUBDVSMT=3 6
  9.0 SUBDVSND=ALL
 10.0 MONVALS=JUN JUL
 11.0 STARTYR=1948
 12.0 LASTYR=1980
 13.0 ANS=YES
 14.0 CHOI1=JUN JUL
 15.0 PICK1="TJUNJUL"
 16.0 COMBINCT=1
 17.0 GET UTREVEND
 18.0 EXECUTE UTREVEND
 19.0 WTYPE=PRCP
 24.0 MONVALS=JUN
 28.0 CHOI1=JUN
 29.0 PICK1="PJUN"
 30.0 COMBINCT=1
 31.0 EXECUTE UTREVEND
 32.0 GET BARBXAP
 33.0 AAA=BARBXAP(INTS(10 42),(MN MT ND))
 34.0 BBB(,7)=SUMROWS(TJUNJUL*AAA)/SUMROWS(AAA)
 35.0 BBB(,6)=SUMROWS(PJUN*AAA)/SUMROWS(AAA)
 36.0 BARCAST=BBB*VFAM(COEFF)
 37.0 BARCAST=INTPART(AFAM(BARCAST)*10+.5)/10
 38.0 TOP="U.S. BARLEY YIELD FORECAST"
 39.0 DAT=DATE
 40.0 WHATIS BARCAST
 40.1 $$
 40.2 $$
```

(BARPROGN concluded)

```
 40.3 TDLGF="WEAA"
 40.4 TDIP=1901001; TDLP=1952001
 40.5 LSTMXDWK=MAX(TDAM(GET "WKMXFLG"))
 40.6 GET ENDWK80
 40.7 WETHBDRY=ENDWK80(LSTMXDWK)
 40.8 TYPE("NOTE:  THIS FORECAST USES WEATHER THROUGH NOAA WEEK" LSTMXDWK)
 40.9 $$
 41.0 $$
 41.5 SPACE 2
 42.0 "*****************************************************************************"
 43.0 TABULATE(INTS(1948 1980) BARCAST:TITLE TOP)
 44.0 SPACE 3
 45.0 "MAJOR ECONOMIC ASSUMPTIONS:"
 46.0 "U.S. HARVESTED ACREAGE OF BARLEY FOR 1980: 7.581 MIL. ACRES"
 47.0 "FERTILIZER PRICE INDEX FOR 1980: 248"
 48.0 SPACE 1
 49.0 "THIS FORECAST MADE ON" DAT "WITH WEATHER AVAILABLE THROUGH" WETHBDRY
 50.0 "*****************************************************************************"
 51.0 SPACE 2
 52.0 $$
 53.0 $$
 54.0 GET BARTK
 55.0 NOROWSTK=NOROWS(BARTK)
 56.0 BARTK(NOROWSTK+1)=LSTMXDWK BARCAST(NOELS(BARCAST))
 57.0 & AFAM(BBB(NOROWS(BBB)))*AFAM(COEFF)
 58.0 IMPACTS=NAMELIST(NOAAWEEK BARCAST CONST BARAH TREND FERTNOW FERTPREV
 59.0 & JUNPRCP JNJLTEMP)
 60.0 LASTCAST=BARTK(NOROWSTK)
 61.0 THISCAST=BARTK(NOROWSTK+1)
 62.0 DIF2MIN1=THISCAST-LASTCAST
 63.0 TABULATE(IMPACTS LASTCAST THISCAST DIF2MIN1)
 64.0 SPACE 2
 65.0 VARVALS=BBB(NOROWS(BBB))
 66.0 VARVALS COEFF
*67.0 KEEP BARTK
 68.0 END
```

Figure 26--Listing of program MCQPROGN

```
EDITING MCQPROGN
PROGRAM
GET MCQARRAY
GET UTREVBEG
EXECUTE UTREVBEG
"ENTER FINAL MONTH CONTAINED IN DESIRED TRUNCATION (APR, MAY, JUN, OR JUL)"
REQUEST(TRUNCMON)
MNTHNAMS=ARRAY(4 3:"APRMAYJUNJUL")
COLSSEG=ARRAY(4 1:0 5 15 30)+ARRAY(1 5:1 2 3 4 5)
WTYPE=PRCP
STATNAMS=MO OH IN IL IA
SUBDVSMO=ALL
SUBDVSOH=ALL
SUBDVSIN=ALL
SUBDVSIL=ALL
SUBDVSIA=ALL
MONVALS=OCT NOV DEC JAN FEB MAR APR MAY JUN JUL
STARTYR=1979
LASTYR=1979
TREND2=LASTYR-1953
ANS=YES
CHOI1=OCT NOV DEC JAN FEB MAR APR
PICK1="OCTAPRP"
CHOI2=MAY
PICK2="MAYP"
CHOI3=JUN
PICK3="JUNP"
CHOI4=JUL
PICK4="JULP"
COMBINCT=4
GET UTREVEND
EXECUTE UTREVEND
IF (TRUNCMON.LT.JUN) GO TO SKIPTMP1
FREEIF ANSWERMT REVERSE
WTYPE=TEMP
MONVALS=JUN JUL
STARTYR=1980
LASTYR=1980
CHOI1=JUN
PICK1="JUNT"
CHOI2=JUL
PICK2="JULT"
COMBINCT=2
EXECUTE UTREVEND
SKIPTMP1:
AUXARRAY=ARRAY(10 5: 1 1 1 1 1 25 25 25 25 25 TREND2 TREND2 TREND2 TREND2 TREND2)
AUXARRAY(4)=OCTAPRP-(19.65 19.68 21.15 18.78 11.96)
AUXARRAY(5)=MAYP-(4.5 3.81 4.11 4.16 4.01)
IF (TRUNCMON.LT.JUN) GO TO SKIPTMP2
```

(MCQPROGN concluded)

```
 43.0 AUXARRAY(6)=JUNP-(4.69 3.91 4.18 4.23 4.79)
 44.0 AUXARRAY(7)=(JUNT-(73.95 70 71.51 72.73 70.32))**2
 45.0 AUXARRAY(8)=JULP-(5.4 5.16 5.2 5.4 5.56)
 46.0 AUXARRAY(9)=JULT-(78.35 73.5 75.12 76.51 74.8)
 47.0 AUXARRAY(10)=AUXARRAY(9)**2
 48.0 SKIPTMP2:
 50.0 MCQCAST=SUMCOLS(AUXARRAY*MCQARRAY(,COLSSEG(TRUNCMON-3)))
 51.0 MCQCAST=INTPART(MCQCAST*10+.5)/10
 52.0 TOP="STATE CORN YIELD FORECASTS FOR 1980"
 53.0 DAT=DATE
 54.0 WHATIS MCQCAST
 54.1 $$
 54.2 $$
 54.3 TDLGF="WEAA"
 54.4 TDIP=1901001; TDLP=1952001
 54.5 LSTMXDWK=MAX(TDAM(GET "WKMXFLG"))
 54.6 GET ENDWK80
 54.7 WETHBDRY=ENDWK80(LSTMXDWK)
 54.8 TYPE("NOTE:  THESE FORECASTS USE WEATHER THROUGH NOAA WEEK" LSTMXDWK)
 54.9 $$
 55.0 $$
 55.1 SPACE 2
 55.5 "*****************************************************************"
 56.0 TABULATE(NAMELIST(MO OH IN IL IA) MCQCAST :TITLE TOP)
 57.0 SPACE 3
 58.0 "THESE FORECASTS MADE ON" DAT "USING" MNTHNAMS(TRUNCMON-3) "TRUNCATION,
 58.1 & WITH WEATHER AVAILABLE THROUGH" WETHBDRY
 58.2 SPACE 1
 59.0 "*****************************************************************"
 60.0 SPACE 2
 61.0 $$
 62.0 $$
 63.0 GET CRNBXAH
 64.0 ACRESHRV=CRNBXAH(NOROWS(CRNBXAH),STATNAMS)
 65.0 PRODUCT=MCQCAST*ACRESHRV
 65.5 PRODUCT=PRODUCT SUM(PRODUCT)
 66.0 TOP2="REGIONAL CORN PRODUCTION FORECAST FOR 1980"
 66.5 "*****************************************************************"
 67.0 TABULATE(NAMELIST(MO OH IN IL IA TOTAL) ACRESHRV PRODUCT:  TITLE TOP2)
 68.0 "*****************************************************************"
*69.0 SPACE 2
 70.0 END
```

Figure 27--Forecast printout from MCQPROGN

NOTE: THESE FORECASTS USE WEATHER THROUGH NOAA WEEK 19

****

STATE CORN YIELD FORECASTS FOR 1980

	MCQCAST
....	....
MO	95.2
OH	97.5
IN	110.6
IL	121.6
IA	120.3

THESE FORECASTS MADE ON July 16, 1980 USING JUL
TRUNCATION, WITH WEATHER AVAILABLE THROUGH JULY 5, 1980

****

****

REGIONAL CORN PRODUCTION FORECAST FOR 1980

	ACRESHRV	PRODUCT
....	....	....
MO	2300	218960
OH	3900	380250
IN	6350	702310
IL	11130	1353408
IA	13200	1587960
TOTAL		4242888

****

Figure 28--Listing of program MIXPROG

```
EDITING MIXPROG
  1.0 PROGRAM
  2.5 LOAD TDAM
  3.0 GET INTMAT78
  4.0 INTMAT=INTMAT78
  4.5 THISYEAR=INTMAT(11,40)
  4.7 FEBDAYS=INTMAT(12,40)
  5.0 GET STATELST
  6.0 COEFFS=KEPTLIST(COEFFS)
  7.0 PNAMLST=KEPTLIST(PNAMLST)
  8.0 TNAMLST=KEPTLIST(TNAMLST)
  9.0 MNLENGTH=31 FEBDAYS 31 30 31 30 31 31 30 31 30 31
 10.0 DIGMAT1=ARRAY(10 2:"01020304050607080910")
 10.5 DIGMATNE=ARRAY(8 2:"0102030506070809")
 11.0 WEAAIPLP=(1900+INPUTWK)*1000+1
 12.0 INTMONTH=MIN(LOCS(INTMAT(,INPUTWK).GE.1))
 13.0 MONTHLOC=INTMONTH+12*(THISYEAR-1931)
 14.0 INTTWIST=INTS(1 24) INTS(26 48) 25
 14.5 $$$$$
 15.0 WETHRTYP="PRCP"
 16.0 WETHRFLG=1
 17.0 ROUNDTYP=10 10**(1+(INTMAT(INTMONTH+1,INPUTWK).EQ.0))
 18.0 FLIPBOT=7 7
 18.4 $$$$$
 19.0 LABLO:
 19.4 $$$$$
 20.0 FACTOR1=INTMAT(INTMONTH,INPUTWK)/MNLENGTH(INTMONTH) INTMAT(
 21.0 & INTMONTH+1,INPUTWK)/MNLENGTH(INTMONTH+1)
 22.0 FACTOR2=(INTMAT(INTMONTH,INPUTWK) INTMAT(INTMONTH+1,INPUTWK))/FLIPBOT
 23.0 NRPREFIX=WETHRTYP(1) "NR"
 24.0 DIGMAT=DIGMAT1
 26.0 SSTARTT=1;SSTOPP=47
```

```
$$$$$
LABL1:
$$$$$
FOR DDXX=SSTARTT SSTOPP
MAINSTAT=INTTWIST(DDXX)
STAT=STATELST(MAINSTAT)
MAINSTAT STAT
NUMDIVS=NOELS(OBJECT("COEFF" STAT))
V1=WETHRTYP "W" STAT
X1=WETHRTYP "1" STAT
NORMMAT=OBJECT(NRPREFIX STAT)
$$$$$
FOR DX=1 NUMDIVS
V111=V1 DIGMAT(DX)
X111=X1 DIGMAT(DX)
TDLGF="WEAA"
TDIP=WEAAIPLP
TDLP=WEAAIPLP
WKDATUM=TDAM(GET V111)
TDLGF="WEAM"
TDIP=0
TDLP=0
VAR=TDAM(GET X111)
TWOMNTHS=VAR((MONTHLOC MONTHLOC+1))-FACTOR1*
& (NORMMAT(INTMONTH,DX) NORMMAT(INTMONTH+1,DX))+FACTOR2*WKDATUM
VAR((MONTHLOC MONTHLOC+1))=INTPART(TWOMNTHS*ROUNDTYP+.5)/ROUNDTYP
TDIP=TDIPX
TDLP=TDLPX
TDAM(PUT X111 VAR)
ENDLOOP DX
$$$$$
SPACENOW
ENDLOOP DDXX
$$$$$
IF (SSTARTT.EQ.48) GOTO LABL2
SSTARTT=48;SSTOPP=48
DIGMAT=DIGMATNE
GOTO LABL1
$$$$$
LABL2:
$$$$$
IF (WETHRFLG.EQ.2) GOTO LABL3
WETHRFLG=2
WETHRTYP="TEMP"
ROUNDTYP=ROUNDTYP/10
FLIPBOT=MNLENGTH(INTMONTH) MNLENGTH(INTMONTH+1)
GOTO LABL0
$$$$$
LABL3:
$$$$$
TDAM END
UNLOAD TDAM
SPACEPEAK
END
```

REFERENCES

1. Climatological Data (previously published under the title Climatic Data for the United States by Sections). U.S. Dept. of Commerce, National Oceanic and Atmospheric Administration, Environmental Data Service, all issues, 1931-82.

2. Cohen, Stanley and Steven C. Pieper. The Speakeasy III Reference Manual (OS/VS version). Speakeasy Computing Corporation (Chicago, Ill.), 1979.

3. McQuigg, James D. Economic Impacts of Weather Variability (report prepared for National Science Foundation). Dept. of Atmospheric Science, Univ. of Missouri-Columbia, Aug. 1975.

4. Nix, James E. "Severe Winter Weather Affects Livestock Industry." Weekly Weather and Crop Bulletin, U.S. Dept. of Commerce and U.S. Dept. of Agriculture, Vol. 66, No. 11, March 13, 1979.

5. Supplementary Help Documents (level Pi). Speakeasy Computing Corporation (Chicago, Ill.), 1982.

6. Thompson, Louis M. "Weather and Technology in the Production of Corn in the U.S. Corn Belt." Agronomy Journal, Vol. 61 (1969), 453-56.

7. __________. "Weather and Technology in the Production of Soybeans in the Central United States." Agronomy Journal, Vol. 62 (1970), 232-36.

8. U. S. Department of Agriculture, Economic Research Service, Data Services Center. T-DAM Data Management System Users Guide, Version 3.00. Jan. 1, 1979.

9. __________, Statistical Reporting Service. Crop Production, 1981 Annual Summary. (Jan. 15, 1982).

10. __________. Weather and Water Allocation Study For The [sic] President and Congress (report prepared pursuant to the Food and Agriculture Act of 1977). 1979.

11. Weiss, Michael D. Fifty Years of U.S. Weather: Monthly Temperature and Precipitation by State and Farm Production Region, 1931-80. ERS Staff Report No. AGES820708, July 1982.

12. __________. "Weather and Crop Yield Analysis System." Proceedings of the 6th Annual Speakeasy Conference (Chicago, Ill., Aug. 17-18, 1978). Argonne National Laboratory, 1978.